MASTER'S THESIS

Multicast Routing in Mobile Adhoc Networks using Source Grouped Flooding

by Karthikeyan Chandrashekar
Advisor: Dr. John S. Baras

CSHCN MS 2003-1
(ISR MS 2003-2)

The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site http://www.isr.umd.edu/CSHCN/

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2003		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Multicast Routing in Mobile Adhoc Networks Using Source Grouped Flooding	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Advanced Research Projects Agency,3701 North Fairfax Drive,Arlington,VA,22203-1714**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT
see report

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**120**	

ABSTRACT

Title of Thesis:	MULTICAST ROUTING IN MOBILE WIRELESS AD HOC NETWORKS USING SOURCE GROUPED FLOODING
Degree candidate:	Karthikeyan Chandrashekar
Degree and year:	Master of Science, 2002
Thesis directed by:	Professor John S. Baras Department of Electrical Engineering

Ad hoc networks are peer to peer, autonomous networks comprised of wireless mobile devices. The ease and speed of deployment of these networks makes them ideal for battlefield communications, disaster recovery and other such applications where fixed infrastructure is not readily available. Limited bandwidth, energy constraints and unpredictable network topologies pose difficult problems for the design of applications for these networks. The last couple of years has seen renewed research in this field. Specifically in unicast and multicast routing and security issues.

In this thesis, we address the multicast routing problem for ad hoc networks. We present a novel multicast routing protocol called the source grouped flooding

protocol. The protocol creates multicast routes between the source and group members based on hop count distance constraints. We also propose a probabilistic data forwarding mechanism to achieve efficient data dissemination. We present simulation results that capture the performance of our protocol against parameters that characterize an ad hoc network. We find that the protocol is robust against topology changes and achieves efficient data distribution.

MULTICAST ROUTING IN MOBILE WIRELESS AD HOC NETWORKS USING SOURCE GROUPED FLOODING

by

Karthikeyan Chandrashekar

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2002

Advisory Committee:

Professor John S. Baras, Chair
Dr. Richard La
Dr. Samrat Bhattacharjee

DEDICATION

To family and friends.

ACKNOWLEDGMENTS

I am grateful to my advisor, Professor John S. Baras for his advice, support and encouragement. I would also like to thank Dr. Richard La and Dr. Samrat Bhattacharjee for agreeing to serve on my committee and review the thesis. I would also like to thank Dinesh Dharamaraju, Gun Akkor, Ayan Roy Chowdhury and Prabha Ramachandran for their valuable suggestions and comments. I would like to thank my house mates Manikandan Ramasamy, Rakesh Bobba, Akshay Naik and Anand Gadre for their encouragement and patience.

Finally, I am grateful for the support of my research work and graduate studies through the following contracts and grants (all with the University of Maryland-College Park); DARPA F3060200020510, DARPA MDA 9720010025, Telcordia 10073169, Lockheed Martin Corporation and the Maryland Industrial Partnership Program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 Mobile Ad hoc NETworks (MANETs)

The DARPA radio packet networks [1, 2] were the first deployed wireless networks. Rapid improvement in the field of mobile computing and wireless communications technology has induced further research in the area of wireless computing. Ad hoc networks (MANETs) are infrastructureless, autonomous networks comprised of wireless mobile computing devices. MANETs [3] are peer to peer networks in which all the nodes in the network have the same capability and communicate with each other without the intervention or need of a centralized access point or base-station. The mobile nodes or devices are equipped with wireless transmitters and receivers. These antennas can be omni-directional resulting in a broadcast medium or highly directional resulting in a point-to-point network. Due to limited transmission range of wireless interfaces, these networks are multi-hop networks

i.e.; a node may have to relay a message through several intermediate nodes to reach the destination. Thus every node is a router as well as a host in a MANET. The arbitrary movement of the nodes in such networks results in highly dynamic or ad hoc topologies. A MANET can thus be considered as a dynamic multi-hop graph. Lower channel capacity of wireless channels as compared to wired links, coupled with effects of interference, fading and noise reduce the effective available bandwidth for communication. Thus bandwidth is at a premium in MANETs. Moreover since the mobile devices are dependent on batteries for their operation, these networks are also energy constrained.

MANETs are attractive as they provide instant network setup without any fixed infrastructure. The ease and speed of deployment of these networks makes them ideal for battlefield communications, disaster recovery, conferencing, electronic classrooms etc.

1.1.2 Multicasting in MANETs

1.1.2.1 Multicast Communication

Multicast communication [4] is a means of achieving one-to-many and many-to-many communication. A source or a set of sources send data to a group of interested receivers. Broadcast is a special case of multicast where all the nodes in the network are interested receivers or group members. Multicasting is an interesting and important communication paradigm as it models several application areas viz

subscription services (news groups, TV, radio), collaboration or conferencing services (eg. virtual conferencing) etc. In an ad hoc environment, hosts generally co-operate as a group to achieve a given task, thus the MANET model is a suitable environment for the multicast paradigm. Also the multicast model improves network utilization through mass data distribution, which is ideal for bandwidth constrained networks like MANETs. Therefore multicast communication is very important in ad hoc networks.

1.1.2.2 Challenges in Ad Hoc Network Multicast Routing

Multicast routing protocols used in static networks like Distance Vector Multicast Routing Protocol (DVMRP) [5], Multicast Open Shortest Path First (MOSPF) [6], Core Based Trees (CBT) [7], and Protocol Independent Multicast (PIM) [8], do not perform well in ad hoc networks due to the dynamic nature of the network. Frequent topology changes results in repeated reconstruction of the multicast trees. Also most of these protocols are dependent on global routing like link state [9] or distance vector [10] routing, which is unstable in a mobile environment resulting in excessive channel overhead and unreliable routes. Two important factors that make MANET multicasting challenging are:

1. Frequent Topology Changes

   All nodes in a MANET are mobile, this means that the topology is dynamic and routes (unicast and multicast) that existed may not exist some time

later due to the movement of the intermediate nodes. The presence of stale or out-dated routes results in huge packet losses. To alleviate this problem, the protocol should be resilient to topology changes. Reactive protocols [11, 12, 13] that try to mend disconnected paths incur huge control overhead and yet packets are lost during the reconstruction phase. Mesh protocols [14, 15] establish redundant paths to combat topology change. These protocols improve data delivery, however incur huge data overhead due to redundant transmission of data. Due to the topology changes, it is difficult to design a multicast routing protocol that is efficient in distributing data as well as in creating and maintaining the multicast structure.

2. MAC Contention and Collision

Medium Access Control (MAC) protocols have been converging in recent times. The most prevalent of these being IEEE 802.11 [16]. CSMA/CA is a contention based scheme, where nodes contend with neighbors to transmit on the channel. The 802.11b protocol also uses an exponential back-off algorithm to avoid repeated contention. A large number of nodes contending for the medium could hamper the effectiveness of higher layer protocols. In most cases, omni-directional antennas are used, resulting in a broadcast medium i.e., all nodes within hearing range will receive the transmitted packet. A broadcast transmission is essentially unreliable as it is not acknowledged. Thus if two nodes transmit at the same time the packets will collide and

therefore all packets are dropped. The problem of two nodes sensing the medium to be free and transmitting at the same time is called the "hidden node" problem. This problem is avoided for unicast transmissions by the use of RTS/CTS packets [16], however this problem is yet to be solved for broadcast transmissions. The "hidden node" problem is illustrated in Figure 1.1. Node B can hear both nodes A and C, while A and C are out of range of each other. When the medium is idle, nodes A and C will sense the medium as free and transmit. Node B will receive two packets, however both the packets will be dropped due to collision. Since in ad hoc networks, nodes are regularly moving, and assuming data is flowing in different directions, the number of hidden nodes could be high. Thus the collision problem makes the broadcast MAC unreliable. An increase in traffic (data or control) could result in increased collision and contention. Thus it is important to design traffic efficient algorithms to limit collisions and channel contention. Also the node density i.e., the denseness of nodes in a particular region of the network will greatly affect the performance of the 802.11 MAC protocol. An efficient multicast routing protocol can alleviate the detrimental effects of the MAC layer drawbacks.

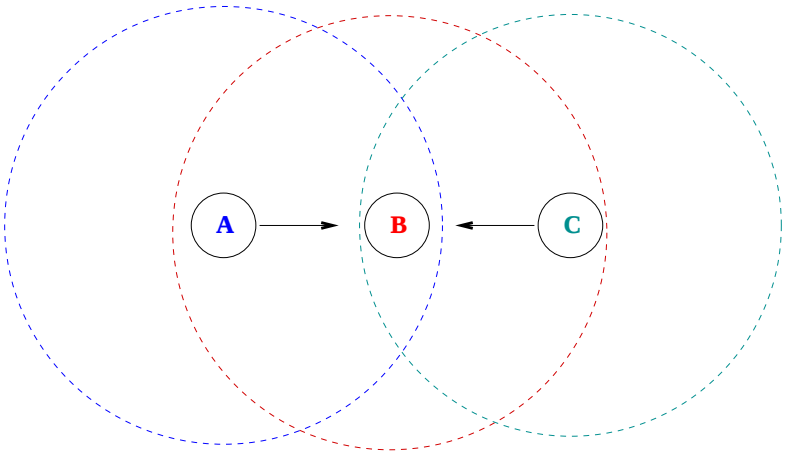

Figure 1.1: Hidden Node Problem

1.2 Contributions

In this thesis we present a novel multicast routing protocol for mobile wireless ad hoc networks. The protocol sets up source groups to the multicast group members based on hop count distance constraints. The mesh structure of the groups provides multiple paths from the source to the group members and hence robustness against node mobility or topology changes. A probabilistic forwarding scheme is developed to improve the efficiency of data distribution. The protocol attempts to achieve the robustness of mesh protocols and improved efficiency through reduced data forwarding. Since the protocol is based on controlled flooding, we expect the protocol to be as effective as flooding in terms of data delivery. The basic protocol and a number of variants were implemented in OPNET [17]. The performance of these protocols was evaluated and compared with the performance of traditional flooding as a multicast protocol. Based on trade-off curves, we suggest a range of

operating values for various parameters that characterize a MANET viz mobility speed, traffic load, network density etc.

1.3 Organization of the Thesis

The rest of the thesis is as follows. In Chapter 2, some of the existing multicast protocols for MANETs are discussed. Chapter 3 introduces the source grouped flooding approach to multicast routing in MANETs. Several improvements are also discussed and four schemes are delineated for performance evaluation. In Chapter 4, the simulation methodology and the results are discussed. Chapter 5 concludes the thesis.

Chapter 2

Review of Related Work

2.1 Multicast Routing Protocols

In wired/static networks, multicast routing protocols, DVMRP [5], MOSPF [6], CBT [7], PIM-SM and PIM-DM [8] are all tree based i.e.; a tree is setup connecting all the members of the multicast group. The difference between these protocols being the approach to creating and maintaining the tree. Multicast routing protocols for ad hoc networks are either tree based or mesh based. Tree based protocols either establish a shortest path tree per source or a shared multicast tree per group connecting the group members. The tree structure is updated reactively as the network topology changes. Some of the existing tree based protocols are [11, 12, 13]. Mesh protocols like [14, 18, 19, 15] create and maintain a mesh structure between the group members i.e., a group of nodes in the network that connect all the group members. This mesh of nodes provides multiple paths to the group members and therefore this structure is more robust against network dynamics due to redundant transmission of data. A comparative study of ad hoc multicast routing protocols

done at UCLA [20] shows that, mesh protocols are more robust to topology changes due to the existence of multiple paths to the destination, however the efficiency is compromised due to redundant data transmissions. Tree based protocols are more efficient than mesh protocols. This is because the shortest path tree is the most efficient data dissemination structure. However, the tree structure is fragile in the face of topology changes. Frequent topology changes could result in massive data loss and excessive control exchange during the tree update process. Mesh based protocols are more reliable for a wide range of mobility speeds. Several multicast protocols like [11, 21, 18, 19] are extensions of unicast protocols. However it is still unclear whether it is a good idea to combine the two functionalities as the problems are sufficiently different to warrant individual attention [22]. In the following sections we introduce some of the existing multicast protocols and their salient features.

2.1.1 Distance Vector Multicast Routing Protocol (DVMRP)

DVMRP [5] is a scheme designed for wired networks though it is also applicable to wireless environments like MANETs. DVMRP uses source flooding of data packets to discover group members. Nodes that are not members will prune themselves from their neighbors. Thus DVMRP creates a shortest path tree between the source and the multicast group members based on the reverse shortest path forwarding mechanism (RPF). Once the tree is setup the data is forwarded by the

nodes on the tree. DVMRP is a soft state protocol and therefore periodic flooding and pruning is carried out to discover new group members. Also new members can explicitly join the tree by sending an explicit *graft* message. Though DVMRP is widely used in Internet multicast routing, it is not suitable for MANETs as the data flooding overhead would be considerable. DVMRP is described here to introduce source initiated or on-demand routing.

2.1.2 On Demand Multicast Routing Protocol (ODMRP)

ODMRP [14, 23, 24] is a mesh based protocol. It creates a mesh of nodes called the *forwarding group* which forward the multicast data packets. Multicast routes are generated only when the sources have data to send and thus this is an on-demand protocol. It maintains soft state information regarding group membership i.e., members don't explicitly join or leave the group. When a source has data to send and it does not have group membership information it floods a JOIN QUERY message. The JOIN QUERY is also periodically broadcasted by each source to the entire network to refresh the group membership information. Every node receiving this packet stores the previous hop node address and rebroadcasts the packet. Thus the intermediate nodes learn the backward or the reverse path. When a member receives the query it creates and broadcasts a JOIN REPLY message that contains the next node information for that source. When a node receives a reply it checks to see if its ID matches with one of the next node IDs in the reply. If a match

exists then the node becomes a part of the *forwarding group*. The node then creates its own reply message containing the next node IDs built upon matched entries. The JOIN REPLY message is thus propagated by each forwarding group member until it reaches the source. This process thus builds a mesh of nodes called the *forwarding group* that connects the sources and the receivers. ODMRP is independent of the unicast routing protocol and in fact it can be used as an unicast protocol. The notion of a *forwarding group* was first introduced in FGMP [25], this protocol creates the *forwarding group* based on receiver advertising and source advertising. FGMP can be considered as a predecessor to ODMRP and thus is not discussed in detail. FGMP is independent of the unicast routing protocol. In both these protocols it is required that the source JOIN QUERY messages are synchronized. This could result in excessive packet loss and possible non optimal routes.

2.1.3 Core-Assisted Mesh Protocol (CAMP)

CAMP [18] attempts to generalize the notion of core-based trees [7] into multicast meshes which would improve connectivity between the group members and the sources. In CAMP each multicast group can have several *core* nodes. The mapping of the group address to core addresses are disseminated from each core out to the network as part of group membership reports. A node wishing to join a multicast group checks if any of its neighbor nodes are group members, if so, the node

announces its membership via a CAMP UPDATE message, otherwise, the node attempts to reach a core node for the group by sending a JOIN REQUEST. If this fails then the node tries to reach a group member through an expanding ring search. CAMP uses a heartbeat mechanism to determine if all the reverse paths exist in the mesh. When a node finds that it is no longer receiving packets from the node in the reverse path to the source, it sends a PUSH JOIN message towards the source to ensure that the mesh contains all reverse shortest paths from all receivers to all sources. The nodes also periodically send broadcast updates to refresh the status of the "relay" nodes, these are nodes that are not part of the multicast group but yet take part in data forwarding. CAMP relies on an unicast protocol that guarantees correct distances to all destinations within finite time e.g. [26].

2.1.4 Flooding as a multicast routing scheme

In flooding each node that receives a packet will re-broadcast that packet and this process continues until all the nodes have received the packets. Flooding is normally used to achieve broadcast in ad hoc networks i.e. when all the nodes in the network need to receive a particular packet. Since broadcast is a specific case of multicast where the number of group members is the same as the number of nodes in the network, flooding can used as a multicast protocol. Flooding is very reliable due to the redundant packet transmissions however the overhead associated with data retransmission would be very high. The performance of flooding as a

multicast protocol has been analyzed as a function of mobility in [27]. It has been shown empirically that even flooding is not reliable at high speeds (100 m/s).

The broadcast storm problem associated with flooding is discussed in [28], here several simple schemes are described to reduce the overhead associated with data re-broadcast. These are summarized below:

- Probability based: here the nodes receiving an original copy of a flooded packet will re-transmit the packet with a predefined probability 'p'.

- Counter based: here when a node receives a packet for the first time it waits for a prespecified period of time to account for duplicates. The node then retransmits the packet if less than 'k' duplicates were received.

- Distance based: here, a node decides to retransmit a packet based on the distance to the node from which it received the packet. Signal strength is used to determine the distance to a node.

- Location based: here the location of a node as provided by GPS is used to determine the extra coverage that a nodes transmission will achieve. Thus in this approach the decision to retransmit a packet is based on extra coverage it would achieve.

The results in the paper [28], show that using one or more of these schemes will reduce the data overhead due to re-broadcasts and also reduce MAC layer contention and collision. These schemes like the basic flooding scheme do not

guarantee data delivery to all nodes.

2.1.5 Neighbor Supporting Ad hoc Multicast Routing (NSMP)

NSMP [15] is again an on demand routing protocol similar to ODMRP. This protocol attempts to reduce the control overhead by limiting the periodic flooding of control information. Initially a source will flood a FLOOD_REQ message to create the path to all members. The creation of the *forwarding group* is exactly the same as in ODMRP. In addition to the forwarding group NSMP also maintains the *neighbor group* which is the set of nodes that are neighbors to *forwarding group* nodes. Periodically every source will send a LOCAL_REQ message to refresh the routes. This message is re-broadcast only by the nodes in the *forwarding group* and *neighbor group*. Thus the overhead due to exchange of control information is reduced, however the drawback here is that, all disconnected paths and network partitions cannot be resolved using local flooding. An elected source called *group leader* will periodically flood a FLOOD_REQ to the entire network to resolve network partitions. As in ODMRP only the nodes in the *forwarding group* will forward the data. Like ODMRP, NSMP does not rely on any unicast routing protocol.

2.1.6 Multicast Core Extraction Distributed Ad hoc Routing (MCEDAR)

MCEDAR [19] is an extension of the CEDAR [29] unicast routing protocol. The protocol uses the CEDAR architecture to determine the dominating set of core nodes for each node in the network. These core nodes determine the *core graph* through advertisements. Using the core graph, a core node is able to set up virtual links to other core nodes. When a node loses its core due to mobility it nominates another dominating node as the core which is added to the graph. MCEDAR creates a subgraph of the *core graph* called the *mgraph*. The mgraph consists of core nodes servicing multicast group members. Thus the process of joining and leaving the multicast group is undertaken by the core node of the group member. Data forwarding is achieved by creating a source based tree on the mgraph. This protocol aims to achieve robustness by creating a mesh multicast group structure and data efficiency using source tree forwarding.

2.1.7 Ad hoc Multicast Routing Protocol (AMRoute)

AMRoute [13] is a tree based protocol that establishes a user level or virtual tree connecting the multicast group members. The protocol maintains a mesh of unicast tunnels connecting the members. The mesh structure is easier to maintain and is more robust. The protocol then periodically creates a tree structure from the mesh for efficient data forwarding. Since the group members are connected via

tunnels, movement of intermediate nodes does not affect the multicast routes. The performance of this protocol is determined by the performance of the underlying unicast protocol. The performance of this protocol is constrained by the creation of loops due to the movement of intermediate nodes forming the tunnel between the nodes.

2.1.8 Ad hoc Multicast Routing Utilizing Increasing ID-numbers (AMRIS)

AMRIS [12] is a tree based protocol. In this protocol nodes belonging to a multicast session are dynamically assigned ID numbers. The ordering between these numbers is used to determine data flow, and the sparseness between these IDs allows for quick connection repairs. The protocol creates a shared multicast tree rooted at a special source node. This special source node is responsible for initiating the generation of the ID numbers. The ID number of a node is directly related to its distance from the special source node. The protocol operates in two phases. In the initialization phase nodes are assigned IDs and the shared tree is created based on these IDs. The tree maintenance phase is responsible for repairs and new member joins.

2.1.9 Multicast Ad hoc On demand Routing protocol (MAODV)

MAODV [11] is an extension of AODV [30]. This protocol creates a shared multicast tree per multicast group. The tree is rooted at a special node called the group leader. The group members explicitly join the existing tree using a request reply phase followed by an activation phase. The group leader periodically advertises sequence numbers to the entire network. The sequence numbers along with the hop count distances enables the nodes to choose shortest loop free paths to the tree. The advertisements also help in reconnecting partitioned trees.

2.1.10 Light weight Adaptive Multicast routing protocol (LAM)

LAM [21] is an extension to TORA [31]. The protocol is tightly coupled with TORA and uses the DAG information provided by TORA to establish a loop free multicast tree rooted at a CORE node. LAM generates limited control overhead as it uses the routing abilities of TORA. The key feature of LAM is that it does not require any timer based messaging. However LAM being a core based protocol suffers from single point of failure and traffic concentration at the core.

Chapter 3

Multicast Routing Using Source Grouped Flooding

In this chapter we describe a new multicast routing protocol for mobile ad hoc wireless networks. The protocol establishes a source based mesh of nodes called the *flooding group* to distribute data for that source. The notion of a flooding group is different from a *forwarding group* described in Chapter 2 in that, the *flooding group* is created based on hop count distance metrics and distance constraints where as the *forwarding group* is created based on the reverse shortest path mechanism. Also the *forwarding group* is a group based mesh of nodes while the *flooding group* is a source based mesh. The protocol aims to improve connectivity and data delivery amidst topology changes and node movement. It avoids the drawbacks of *tree based* protocols in ad hoc networks viz fragility against topology changes, non-optimal paths in the case of shared trees, tree partitions, frequent tree reconstruction etc. Also the protocol avoids excessive redundant data transmission due to multiple paths by using *probabilistic data forwarding*. Thus this

protocol attempts to combine the robustness of the mesh structure by establishing multiple paths and improved efficiency by using a probabilistic data forwarding. This is an on-demand protocol i.e.; control messages are distributed only when the source has data to send, thereby reducing channel overhead. The protocol uses a *soft-state* approach to maintain multicast group membership. The members do not send explicit messages to leave the group. The protocol is independent of the underlying unicast routing protocol.

3.1 Creation of the Flooding Group

In this protocol, each source creates routes to the multicast group members *on demand*. A request phase initiated by the source followed by a reply phase by the group members results in the formation of the *flooding group* for that source. The following distance constraints determines the formation of the flooding group for a source.

$$D_{sn} \leq D_{sm} \tag{3.1}$$

$$D_{mn} \leq D_{sm} \tag{3.2}$$

where, D_{sn} is the hop count distance between the source and the intermediate node. D_{sm} is the hop count distance between the source and the multicast group member. D_{mn} is the hop count distance between the multicast member and the intermediate node.

Nodes in the network use distance constraint 3.1 as a decision criteria to join the flooding group. The nodes learn these distance metrics during the request-reply phase. A source, while it has packets to send, periodically broadcasts a JOIN REQUEST message. The JOIN REQUEST message contains the *multicast group address* and a *hop count* field. This periodic broadcast refreshes the *flooding group* as follows. When a node receives a non-duplicate JOIN REQUEST, it stores the *hop count* for that source (D_{sn}) and re-broadcasts the packet after incrementing the hop count. A multicast group member upon receiving a JOIN REQUEST, stores hop count distance to source D_{sm}, waits for a short fixed interval and then broadcasts a JOIN REPLY message. The delay ensures that the JOIN_REQUEST has propagated past the member. The JOIN REPLY contains the multicast group information and the hop count to the source. The TTL (Time To Live field in the IP header) for this message is set to the hop count from the source (D_{sm}). This ensures that the reply message does not propagate beyond the source. When a node receives a JOIN REPLY the node will compare its stored hop count to the source (stored during the request phase), and the value in the *hop count* field of the reply message. If the hop count distance constraint 3.1 is satisfied i.e., the node's stored hop count to the source is lesser or equal to the hop count value in the packet, the node becomes a flooding node for this source. If the stored hop count is greater, then the packet is dropped. The propagation of the reply message is limited by distance constraint 3.2. Only nodes that are activated as flooding nodes, propagate the reply message. Moreover, the node re-broadcasts the reply

message only if it is not activated as a flooding node during the current route refresh sequence. Therefore a node will re-broadcast only the first reply message for each source during a particular refresh sequence. The protocol thus creates a flooding group for each source consisting of nodes that satisfy hop count distance constraint 3.1; the set of nodes being determined by 3.2. Each source thus creates its own *flooding group*, connecting the source to all the group members. The source maintains a different *flooding group* for each multicast group, as the group membership is different for different groups.

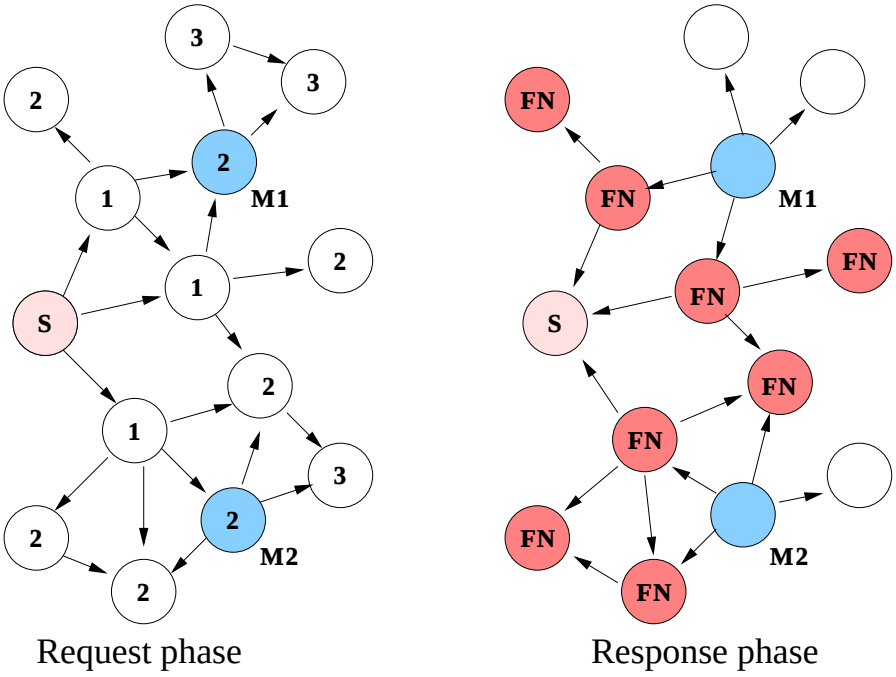

Figure 3.1: Flooding Group Formation

Figure 3.1 illustrates the flooding group creation process for a simple case of one source and two multicast group members. In the request phase the source S

broadcasts the JOIN REQUEST to all nodes in the network. The arrow flow shown in the request phase gives an idea about the propagation of the request message and does not capture every packet transmission. When a node receives a request packet it stores the hop count value in the packet. In the figure, the numeric value in each node denotes the hop count distance from the source. The multicast group members M1 and M2 in the figure receive the request in two hops. Hence they respond with a JOIN REPLY message with the TTL set to 2, which is the hop count distance from the source. The response phase depicts the exact message exchange between the nodes. Nodes for which distance constraint 3.1 is satisfied upon receiving the reply will become flooding nodes, these are marked as FN in the figure. It is seen that only flooding nodes will re-broadcast the reply message provided the TTL is not zero. Also nodes which have stored hop count distance to the source greater than what the member advertised will not re-broadcast the reply message.

3.2 Flooding Group Update and Soft-State

The periodic re-broadcast of the JOIN REQUEST message will reinforce the *flooding group* for each source. This route refresh accounts for topology changes due to mobility and discovery of new members. The protocol maintains soft-state information regarding the group members. A member does not send explicit messages to join or leave the group. When a member joins the group, it starts responding

to JOIN REQUEST messages; to leave the group it simply stops responding to JOIN REQUEST messages. The nodes in the *flooding group* that are not refreshed during the request-reply phase will timeout and will be purged from the group.

3.3 Detection of Duplicate Packets

Each source includes a *broadcast sequence number* in every packet. The broadcast sequence number is a combination of the source address and a counter value and uniquely identifies each packet generated by the source. When any node receives a packet which has a sequence number greater than the stored value, the packet is processed and the cache is updated. If not the packet is a duplicate and is dropped.

3.4 Data Forwarding

Once the *flooding group* is created, a source has fresh, active routes to all multicast group members. When a source transmits a data packet, only the nodes in the flooding group for that source will forward the data packet. All duplicate data packets identified using the unique source broadcast id are dropped.

Figure 3.2 illustrates the data forwarding mechanism and the possible problems arising due to redundant data transmissions. Steps 1 to 4 delineate the data forwarding mechanism. In step 1 the source S transmits the data packet, another network node CN simultaneously transmits another packet as it is 'hidden' from S.

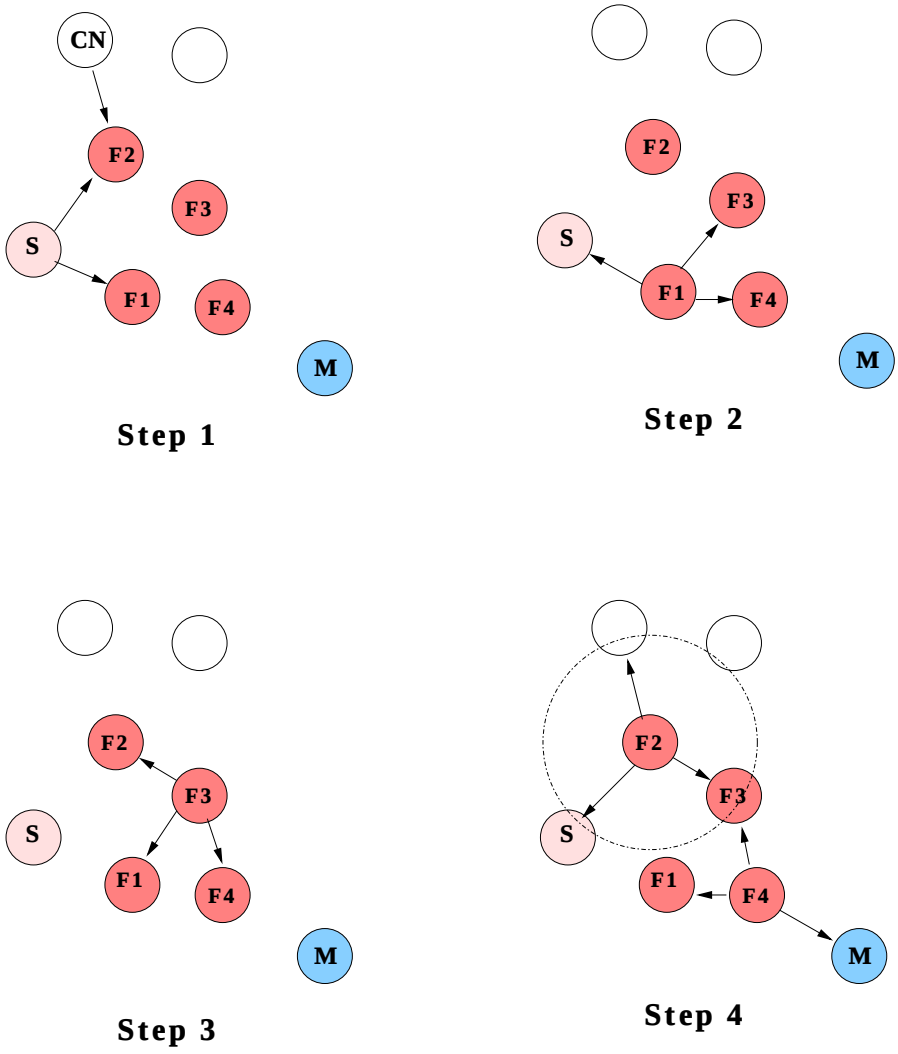

Figure 3.2: Contention and Collision during Data Forwarding

Hence flooding node F2 faces a collision and therefore both packets are dropped. Node F1 forwards the packet received from S, this is received by F3 as shown in step 2. In Step 3, F3 transmits the data packet which node F2 receives as an original packet. In step 4 we see that F4 has transmitted the packet towards the member, however node F2 retransmits the packet in a region of the network through which the data packet has already passed. This transmission by F2 results in additional channel usage, contention and collision. Also this additional transmission may not improve data delivery.

3.5 Hop Count Data Forwarding

To reduce MAC layer contention and collision due to redundant transmission of data, we have included a *hop count* field in the data packet. When the source sends the packet, this field is initialized to zero. When an active flooding node receives a data packet, it compares its latest hop count value for this source with the hop count field in the data packet. The node re-broadcasts the packet only if the stored hop count is greater than the hop count value in the packet. The node stores its hop count distance to the source in the data packet before retransmitting it. This mechanism ensures that data packets are not repeatedly transmitted in the same region of the network and allows the flooding wave to progress effectively. We expect this feature to reduce protocol overhead and also alleviate contention and collision effects at the MAC layer.

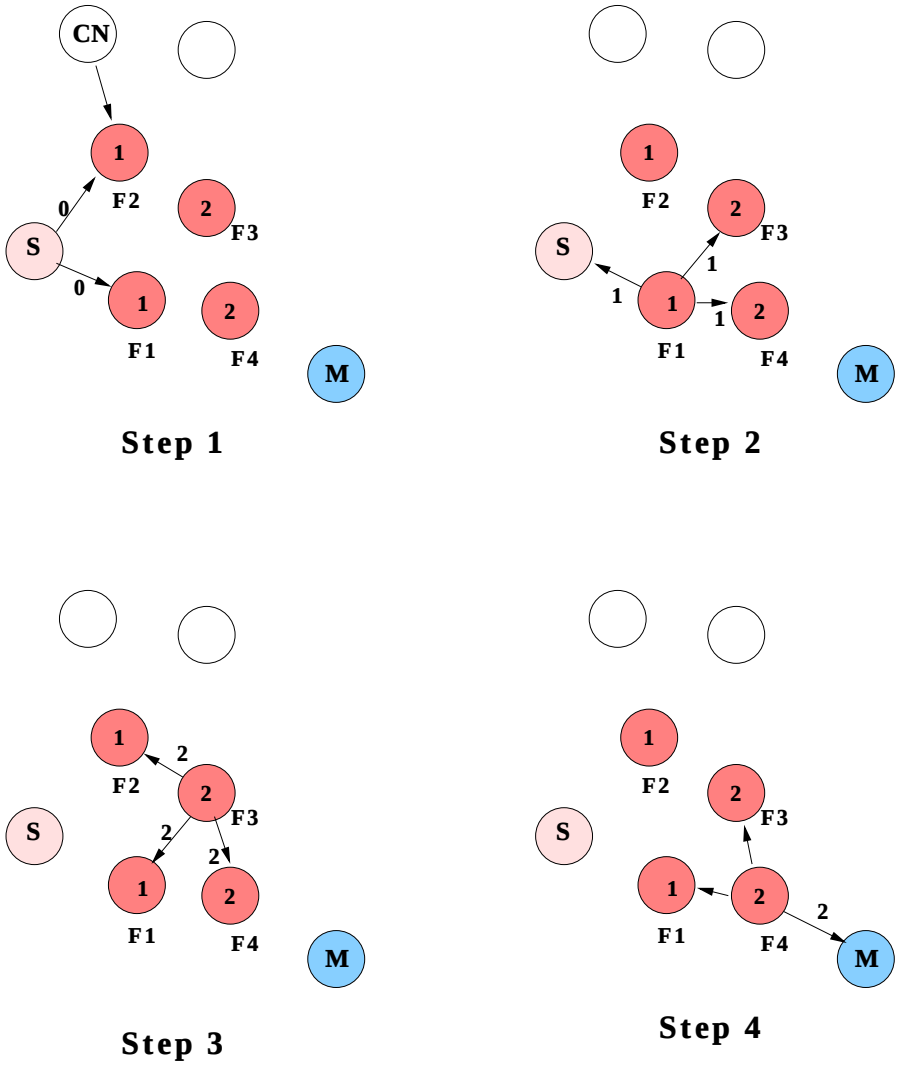

Figure 3.3: Hop count based Data Forwarding

Figure 3.3 illustrates the enhanced data forwarding mechanism. The numeric values inside the nodes represent the hop count distances as learned during the request-reply control phase. The arrows indicate the destinations that a particular transmission will reach and the value on the arrows indicate the hop count value in the data packet. Steps 1 to 3 are the same as in figure 3.2. In step 4 we see that node F2 will receive a packet from F3 with hop count set to 2. In this scheme node F2 will not forward the packet even though it is an original packet as its stored hop count is smaller than that in the data packet. Thus the extra transmission that leads to additional contention and possible collision is avoided.

3.6 Controlling the Size of the Flooding Group

The basic scheme described in Section 3.1, creates a *flooding group* whose size is determined by the hop count distance between the source and the group members, especially the most distant group member. The size could be large if the source and the group members are well dispersed, which is likely in dynamic mobile networks. In this section we present a more strict distance constraint that ensures that only nodes that form the shortest paths between the source and a member will become flooding nodes. The following constraint is derived using the fact that a node lies in the shortest path between a source and a member if the sum of the node's distance to the source and the node's distance to the member is less than or equal

to the distance between the source and the member.

$$D_{sn} + (D_{sm} - TTL_{rep}) \leq D_{sm} \longleftrightarrow D_{sn} \leq TTL_{rep} \tag{3.3}$$

where, D_{sn}, D_{sm}, D_{mn} are as defined for constraints 3.1 and 3.2. TTL_{rep} is the decremented value of the TTL field in the reply message.

D_{sm} is the initial value of the TTL in the reply message sent by the member, and TTL_{rep} is the decremented value of TTL in the reply message that the node receives. Thus (D_{sm} - TTL_{rep}) is the hop count distance between the node and the group member. The nodes use the reduced form of this constraint to decide to join the flooding group. Thus in this modified protocol, when a node receives the JOIN REPLY message sent by a group member, it compares the stored hop count value from the source (D_{sn}) with the TTL_{rep} value obtained from the reply message. The node becomes a flooding node for this source if its hop count value is lesser than or equal to the TTL_{rep} value. The node then modifies the packet to reflect the new TTL value and re-broadcasts it. Only nodes satisfying distance constraint 3.3 can become flooding nodes and these nodes by definition form the shortest paths between the source and the member. As in Section 3.1 the propagation of the reply messages is controlled by distance constraint 3.2. This scheme thus creates a source flooding group consisting of all nodes that form the shortest path between the source and the multicast group members. If multiple shortest paths exist then all nodes in these paths are included in the flooding group. This improvement reduces the number of redundant data packets transmitted and limits the propagation of

the member JOIN REPLY messages.

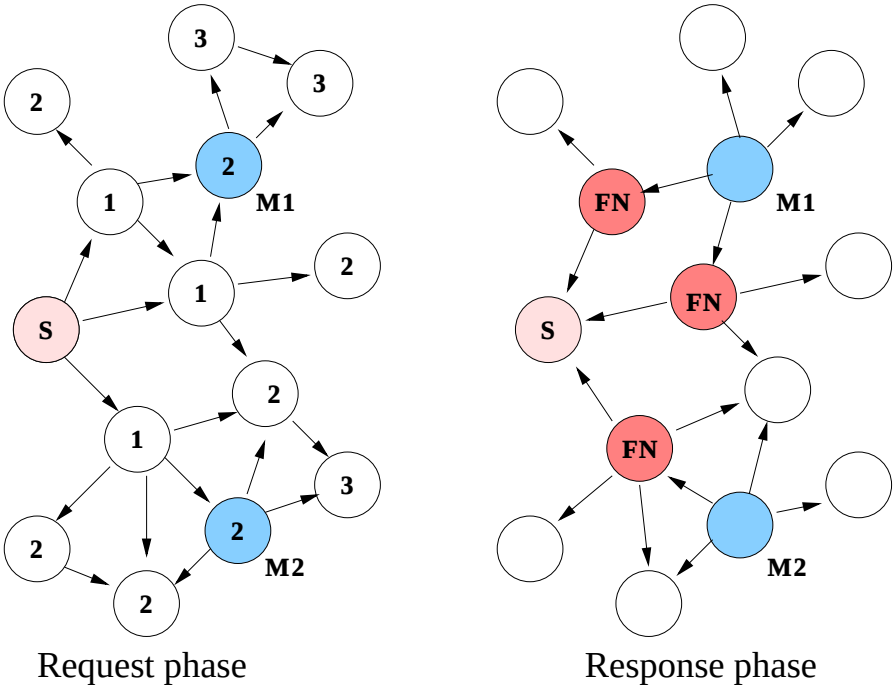

Figure 3.4: Creation of Controlled Flooding Group

Figure 3.4 illustrates the formation of the controlled (shortest path) flooding group. The request phase is the same as in figure 3.1 where both group members M1 and M2 receive the JOIN REQUEST message in two hops. In the response phase both members respond with JOIN REPLY messages with TTL set to 2. Nodes for which the strict distance constraint 3.3 is satisfied will become flooding nodes. It is clear from the figure, that the strict distance constraint used in this scheme optimizes the size of the flooding group such that, only nodes that form the shortest paths between the source and the members become flooding nodes. In this simple case we see that the flooding group consists of 3 nodes, where as in Figure 3.1 the flooding group has 8 nodes.

3.7 Probabilistic Data Forwarding

The *flooding group* provides multiple paths from the source to the group members. Redundant transmission of data along these paths will improve data delivery, however it will result in excessive overhead and also degrade MAC layer performance by contributing to contention and collisions. In this section we present a *probabilistic data forwarding* mechanism to reduce data overhead and describe a method to determine a meaningful value for the retransmission probability (P_{send}) of a packet. In this scheme, when a node receives a non-duplicate data packet, it stores the packet and waits for a short random interval of time for arrival of duplicate packets. The node increments a counter for every data packet received from a node in its peer distance level from the source, i.e., data packets having hop count value same as this node's stored hop count value. All other duplicate data packets are dropped. When the wait interval is over, the node calculates the retransmission probability of the packet using (3.4). The node decides to retransmit the packet with probability P_{send} and drop the packet with probability $(1 - P_{send})$. Once a data packet has been retransmitted, all duplicates irrespective of hop count value will be dropped.

$$P_{send} = \frac{1}{1+n} \tag{3.4}$$

where, n is the number of duplicate packets received from the same hop count peer level.

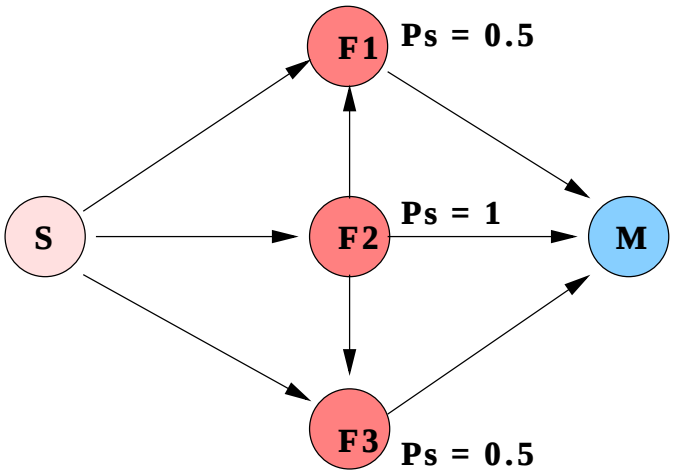

Figure 3.5: Probabilistic Forwarding of data

Figure 3.5 demonstrates the benefit of the probabilistic forwarding scheme. Source S is connected to member M through flooding nodes F1, F2 and F3 that form the shortest paths between S and M. When the source S transmits a packet, F1, F2, and F3 receive the packet. Let's assume, node F2 times out first and transmits with probability 1. Nodes F1 and F3 which are in the same peer hop count level will increment their duplicate counters upon receiving the packet from F2. Thus F3 and F1 will retransmit the packet with probability 0.5. Thus the number of retransmissions is potentially reduced and at the same time, at least one packet is forwarded in each peer hop count level ensuring that the member receives the packet.

A uniform distribution in 'n+1' is used to determine the retransmission probability of a packet. When n is zero i.e., no duplicates are received, the packet is transmitted with probability $P_{send} = 1$. The probability P_{send} here is a measure

of the effectiveness of re-broadcasting a packet and is adaptively derived from the network per packet. The probabilistic nature of the scheme however hampers the reliability of data delivery. Figure 3.6 delineates the drawback of the probabilistic

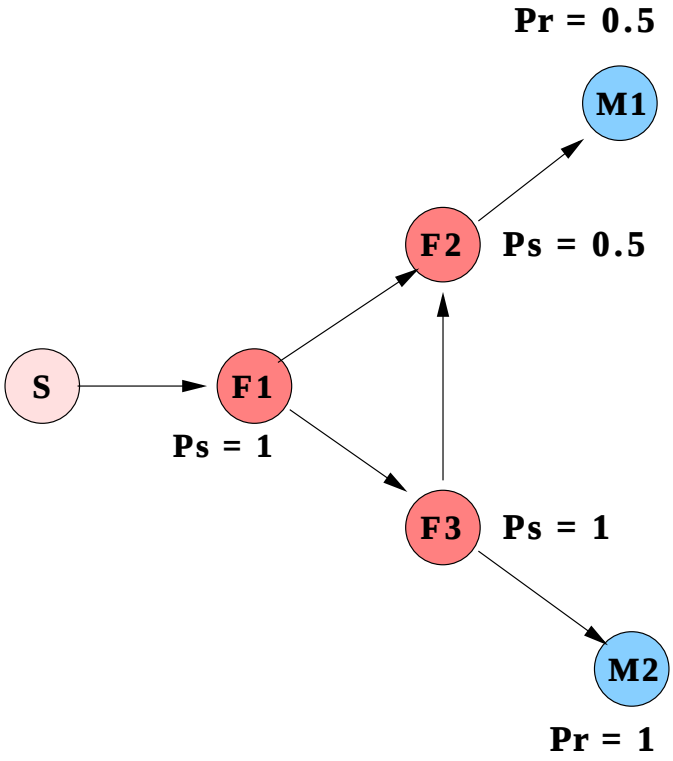

Figure 3.6: Non Guaranteed delivery of data

data forwarding scheme. The source S has paths to members M1 and M2. Flooding nodes F1, F2 and F3 form the flooding group. When node F1 receives the packet from S, it waits for the predefined interval and transmits with $P_{send} = 1$ as it does not receive any duplicates. This packet is received by nodes F2 and F3. Lets assume F3 has a shorter wait period, it then transmits the packet with probability 1. When F2 receives this packet sent by F3 it increments n, which is

the number of duplicates, to 1. Thus when F2 times out, it sends the packet with $P_{send} = 0.5$. Thus member M2 will receive the packet with probability 1 while member M1 receives the packet with probability 0.5. This simple scenario shows that probabilistic forwarding could affect reliability, more so, when the decision by a node to, not retransmit a packet results in network partitions.

3.8 Protocol Timers

The protocol relies on the following timers.

3.8.1 Route Refresh Interval

The refresh interval is a configurable attribute of the protocol. This interval determines the frequency of the route refresh messages i.e. the JOIN REQUEST messages. A source after setting up the flooding group, refreshes the group after every route refresh interval. Since the flooding groups are source based, the JOIN REQUEST messages of the different sources need not be synchronized in time. Ideally this interval should be adaptive to the network environment, varying inversely with mobility. Smaller refresh intervals ensure fresh routes, however resulting in more number of packets and network congestion. On the other hand large refresh intervals may result in outdated group information, thereby affecting data delivery. In our approach since the *flooding group* consists of atleast all the nodes on the shortest paths between the source and the members, we envisage that mobility will

not have a drastic effect on our protocols. A related timer is the *flooding group* active timer. This timer is the time out interval for a particular flooding node i.e., if the route has not been activated in the last active timer period then the node is de-activated as a flooding node and removed from the flooding group. This timer is set to twice the value of the refresh interval so that a node is de-activated only when it is not re-activated in the next two refresh rounds. This prevents accidental de-activation of a flooding node due to loss of refresh messages.

3.8.2 Data Wait Interval

This timer is used in the probabilistic scheme and determines the wait period for duplicates of a particular packet. This timer is composed of two values. A fixed small value to account for propagation delay and queuing delay of packets in the lower hop count level, and a random wait period to ensure that the nodes in the peer hop count level do not transmit data packets at the same time.

3.9 Data Structures

The nodes in the network have to maintain limited state for effective functioning of the protocol. The following tables are maintained by the nodes in the network.

3.9.1 Group Information Table

Each group member stores multicast group information in this table. Each entry contains the multicast group address for which the node is a group member. An entry is created when the node joins a multicast group and deleted when the node leaves the group. When an entry exists for a particular group, the member responds to JOIN REQUESTS for that group.

3.9.2 Flooding Node Table

Every node in the network maintains this table akin to a routing table. Since flooding groups are generated per source, each entry is identified by a combination of the source address and the multicast group address. Each entry contains the Flooding Node flag, activation time and hop count distance from the source. When a node becomes a flooding node it activates the Flooding Node flag and sets the activation time to the current time. Hop count field is updated with every refresh message from the source.

3.9.3 Data Packet Cache

This cache is required only for the probabilistic data forwarding mechanism. Here, every node temporarily stores the original data packet in this cache while waiting for duplicates. It also stores the number of duplicates along with the packet. When the timeout occurs the node reads the corresponding entry in the cache and

probabilistically determines whether to transmit the packet or not. Cache entries are temporary and are removed upon timeout for the packet.

3.10 Algorithms for Evaluation

In this section we specify four algorithms for evaluation. These algorithms are combinations of the schemes and enhancements described in the previous sections.

3.10.1 Basic Source Grouped Flooding Protocol

This algorithm generates the flooding group using distance constraint 3.1 in Section 3.1. The flowchart below represents the algorithm and specifies the schemes used for creation of the flooding group and data forwarding.

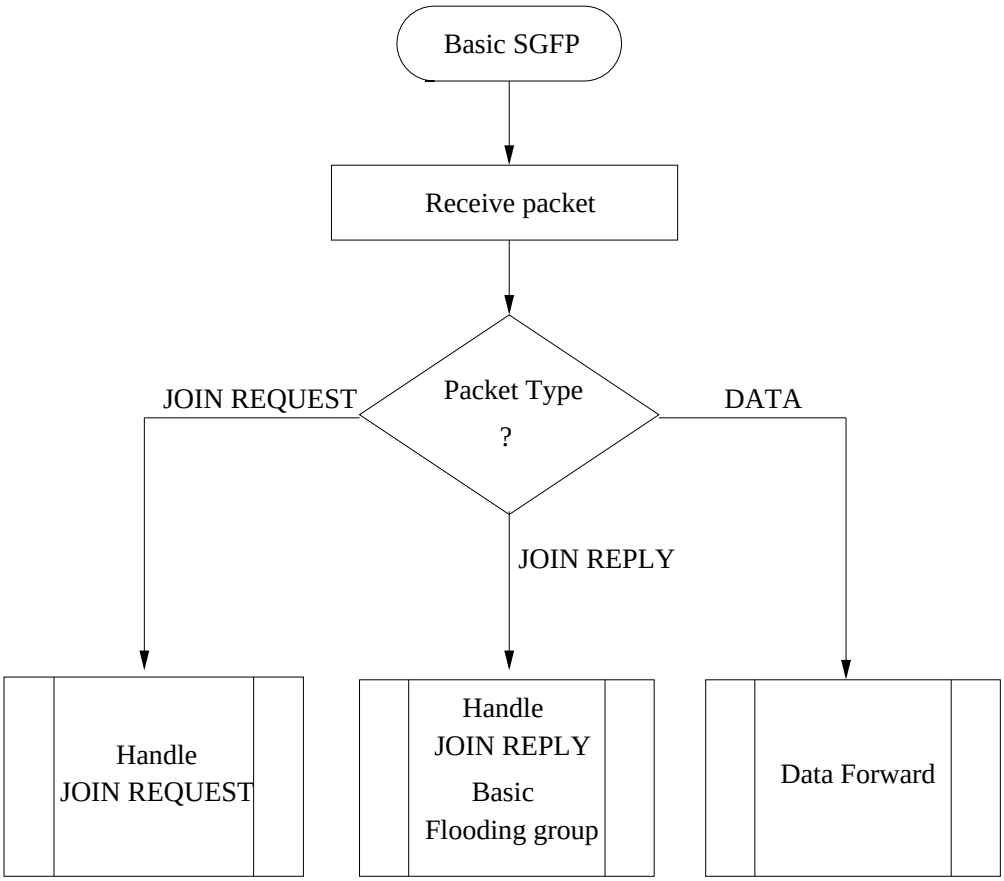

Figure 3.7: Basic Source Grouped Flooding Protocol

3.10.2 Shortest Path Source Grouped Flooding Protocol

This algorithm generates the shortest path flooding groups using distance constraint 3.3. This algorithm is used to evaluate the benefit of creating shortest path flooding groups.

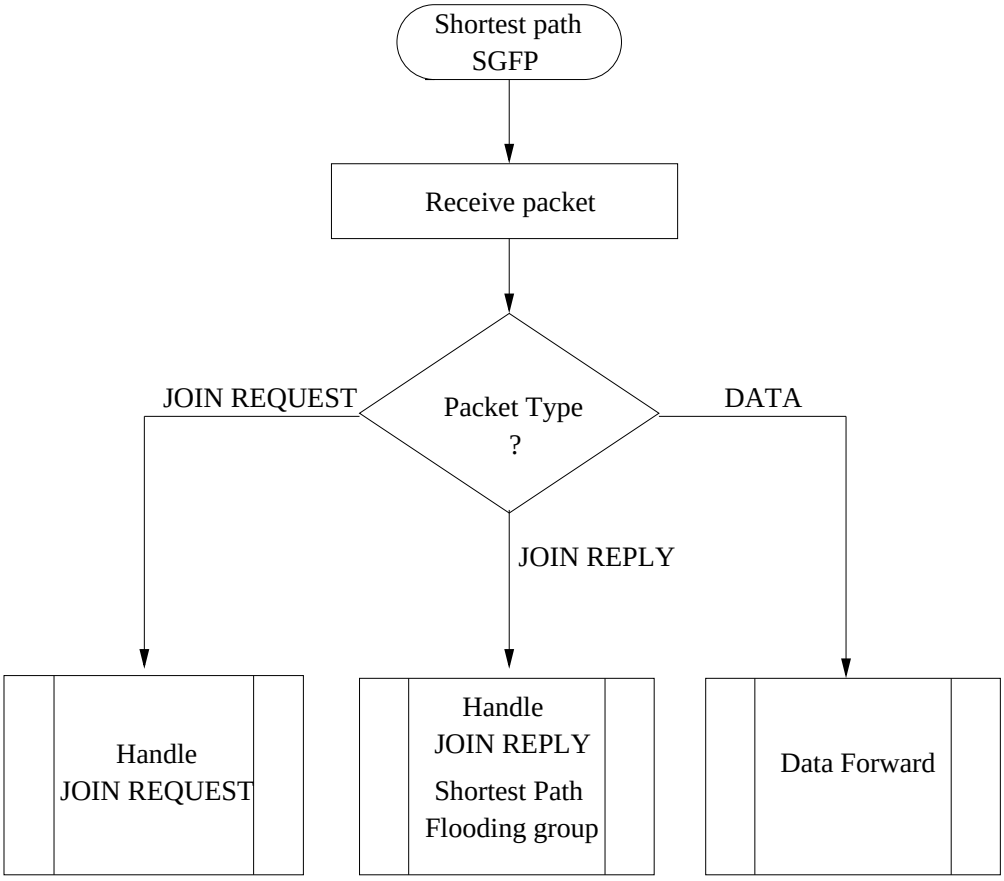

Figure 3.8: Shortest Path Source Grouped Flooding Protocol

3.10.3 Probabilistic Basic Source Grouped Flooding Protocol

This algorithm uses the probabilistic data forwarding mechanism on the basic flooding group.

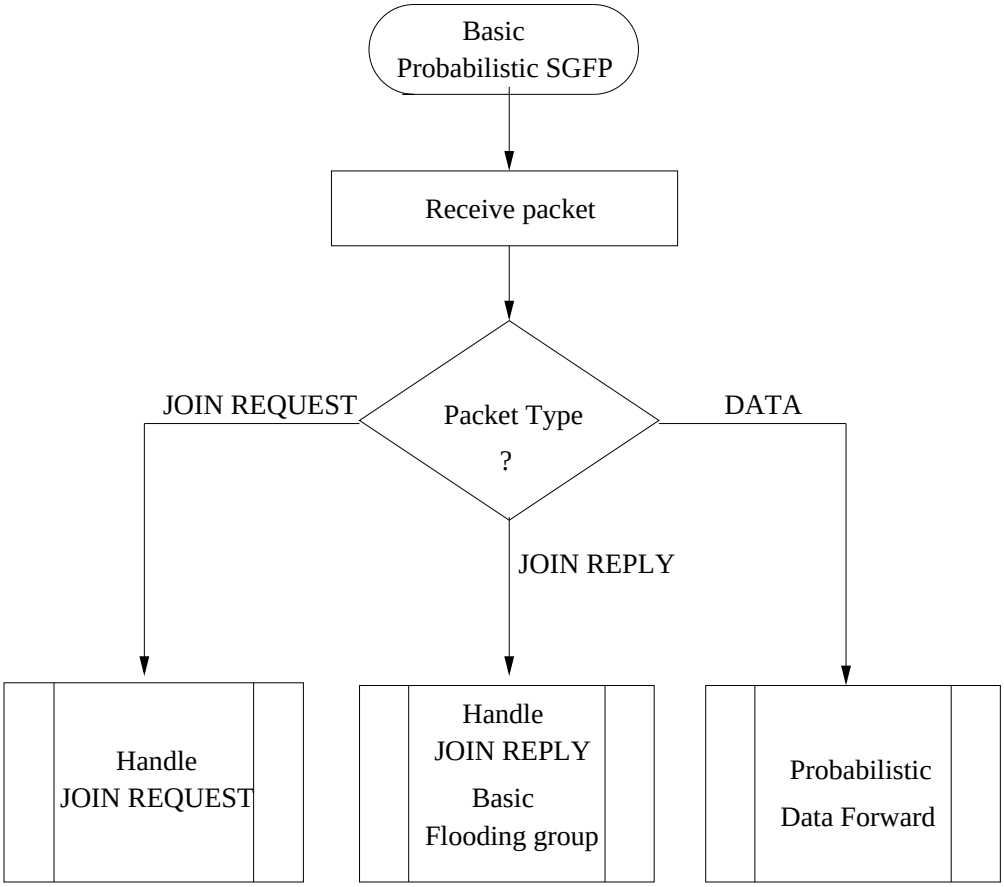

Figure 3.9: Probabilistic Basic Source Grouped Flooding Protocol

3.10.4 Probabilistic Shortest Path Source Grouped Flooding Protocol

This algorithm uses the probabilistic data forwarding mechanism on the shortest path flooding group. This is a comprehensive algorithm using all the enhancements described in the previous sections. This is the primary scheme being evaluated. We expect this scheme to be robust and efficient.

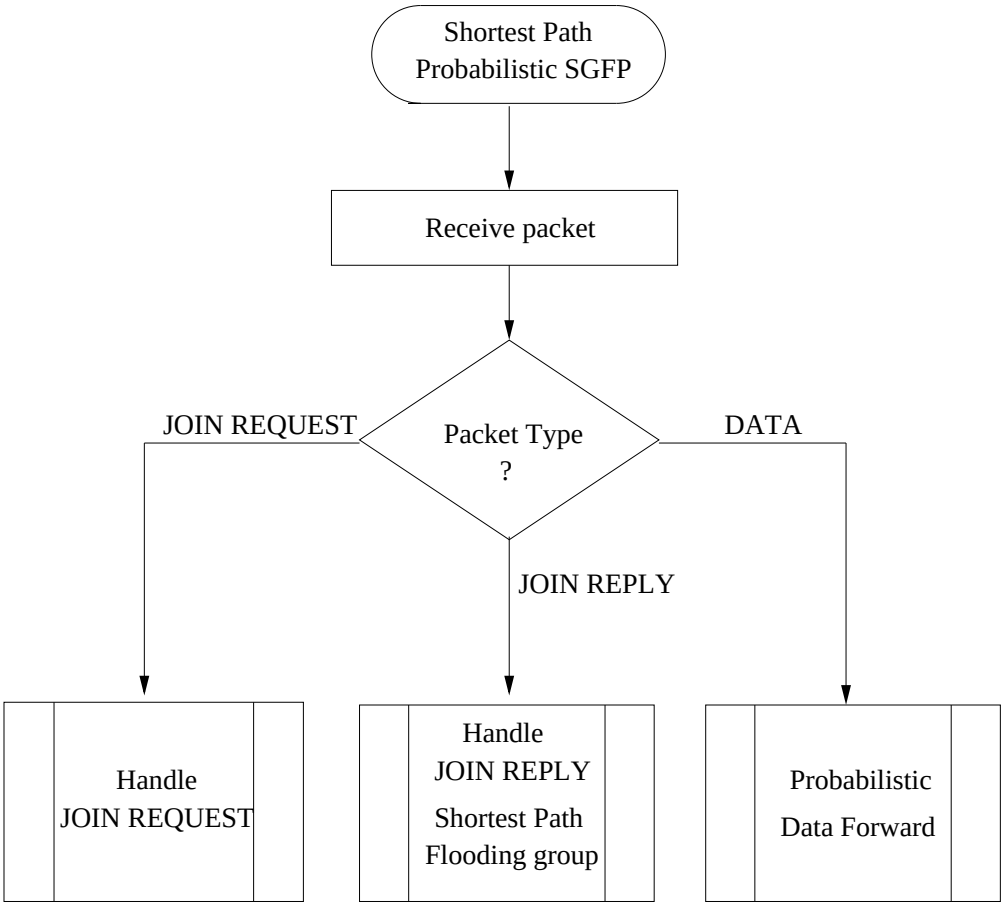

Figure 3.10: Probabilistic Shortest Path Source Grouped Flooding Protocol

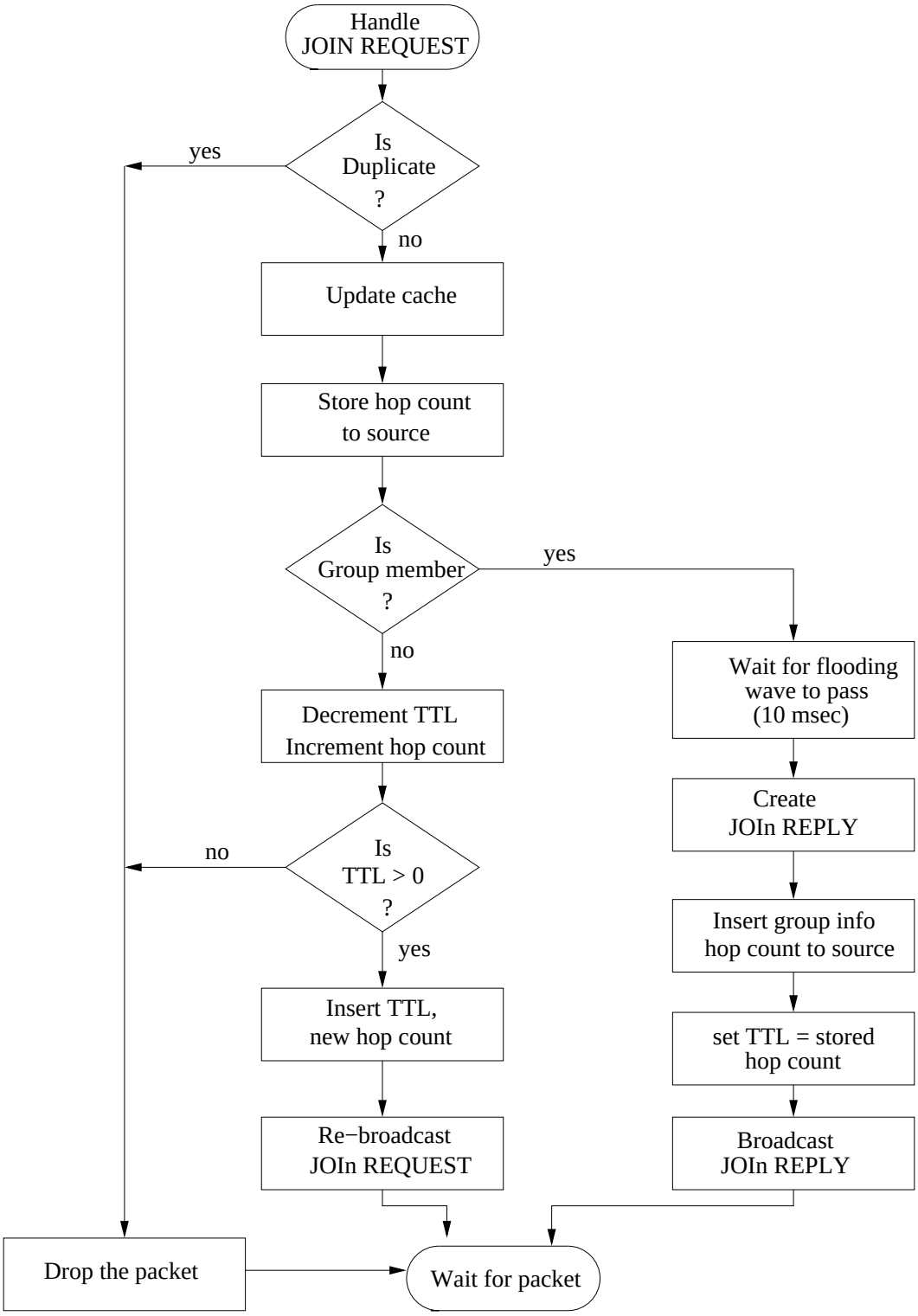

Figure 3.11: Procedure to handle JOIN REQUESTs

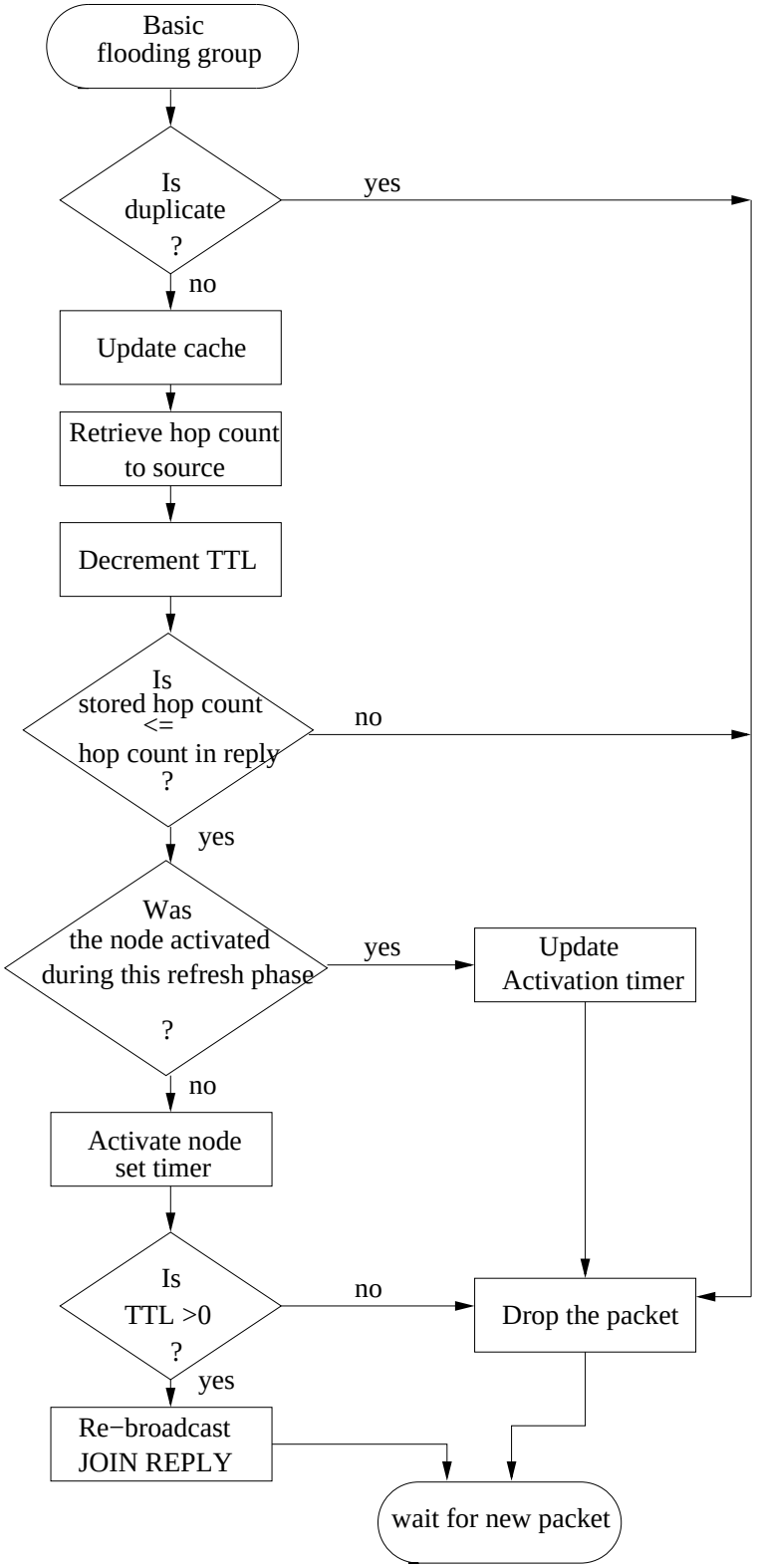

Figure 3.12: Response phase and generation of flooding group

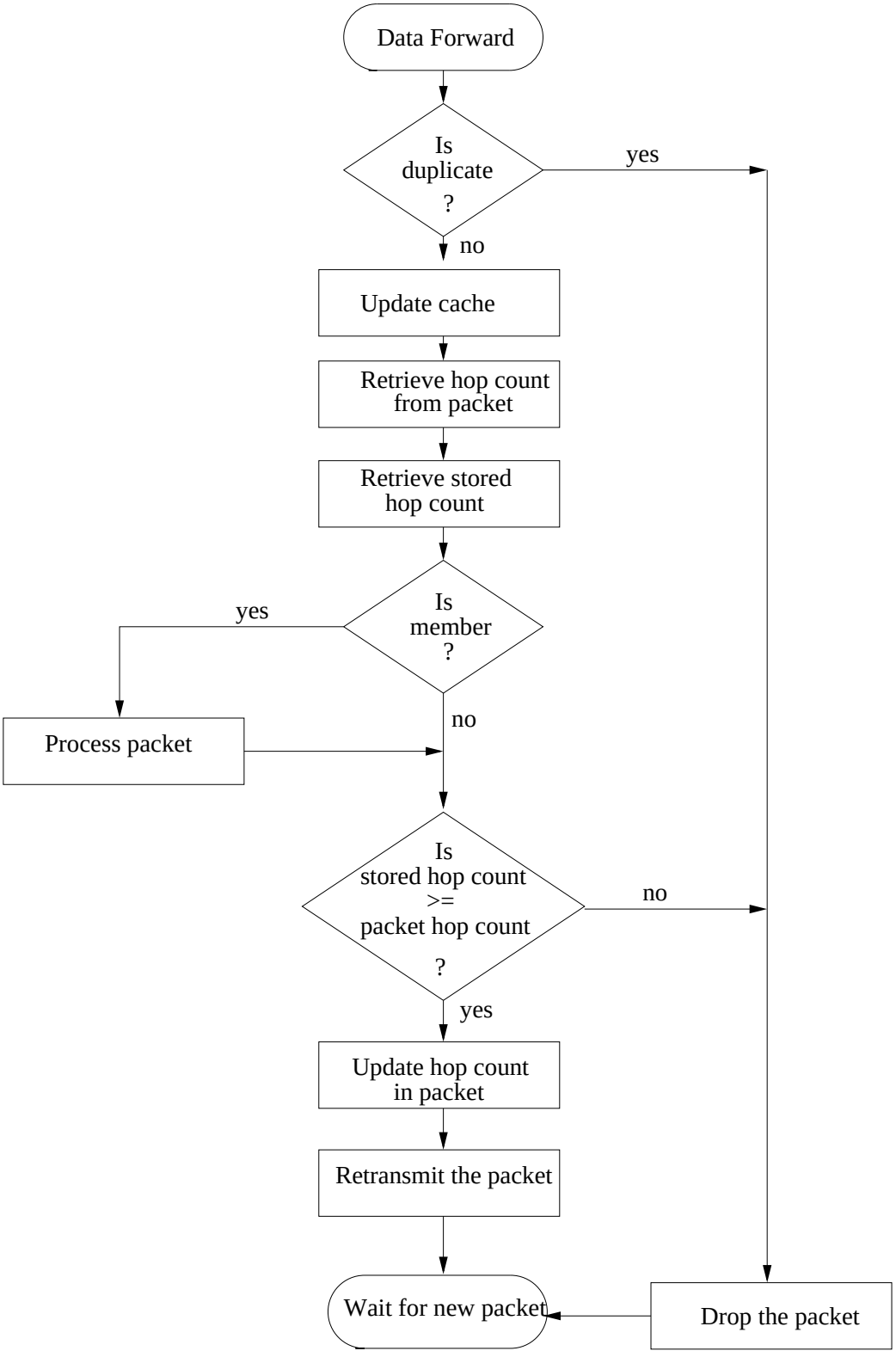

Figure 3.13: Data forwarding procedure

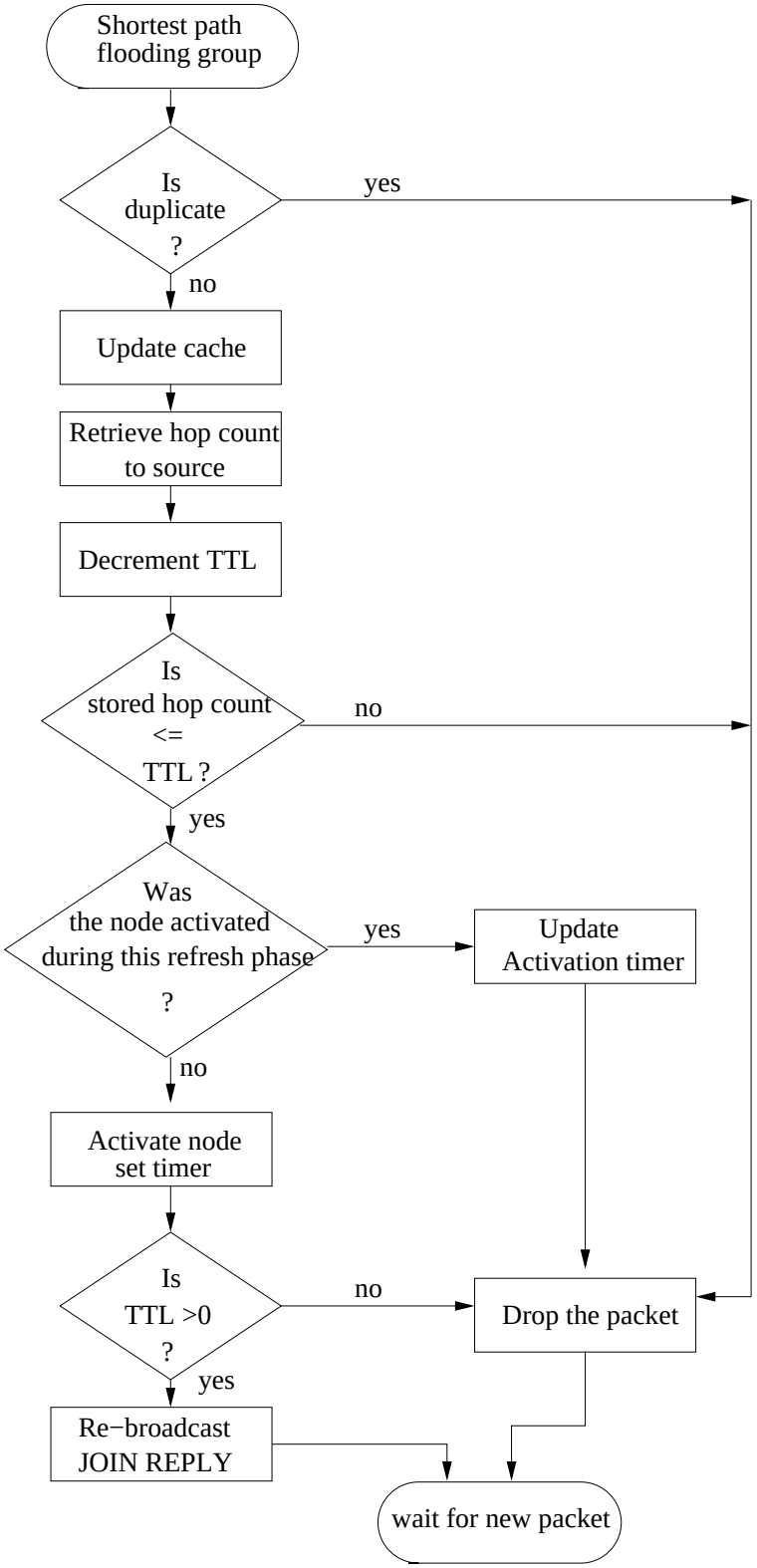

Figure 3.14: Response phase and generation of shortest path flooding group

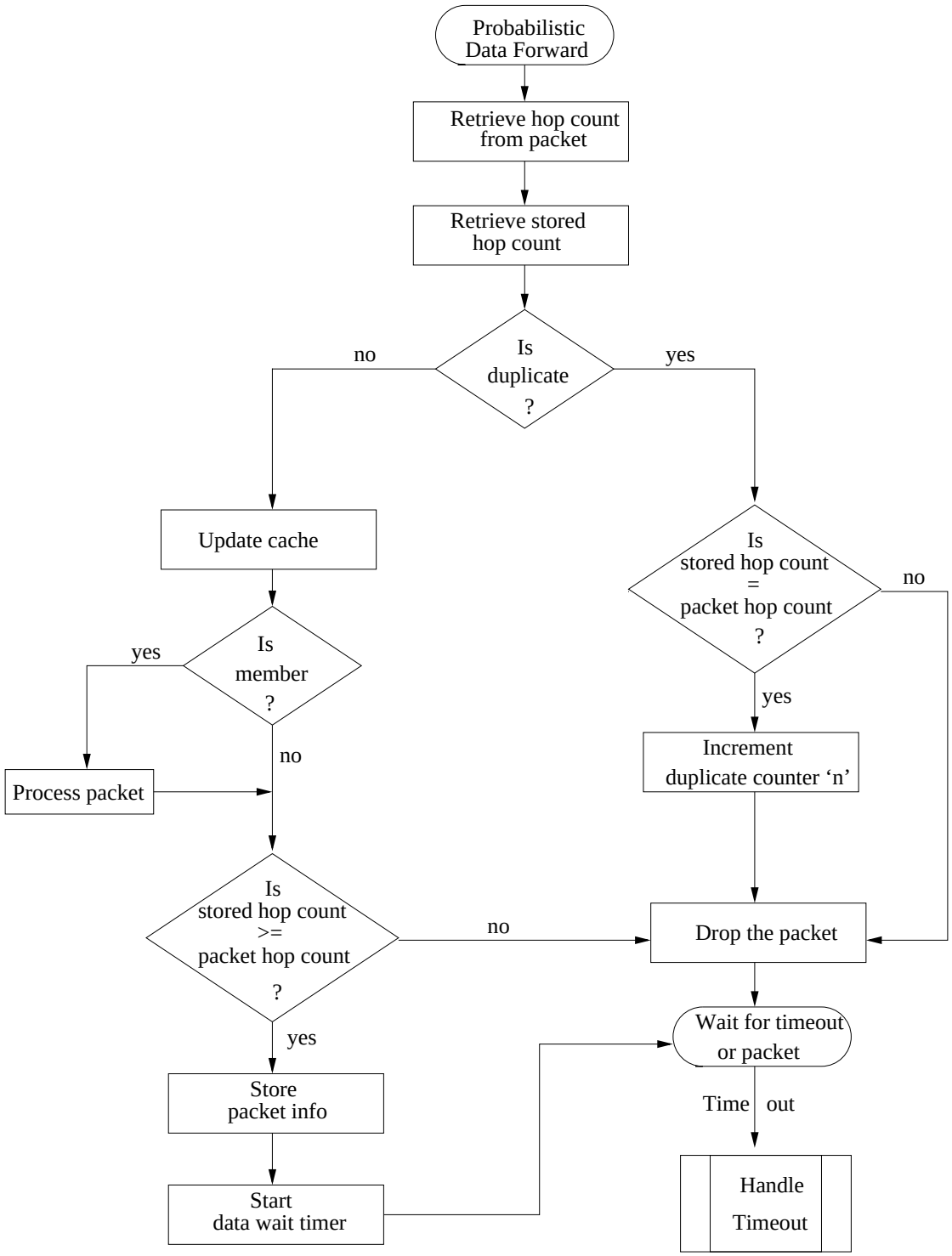

Figure 3.15: Probabilistic data forwarding procedure

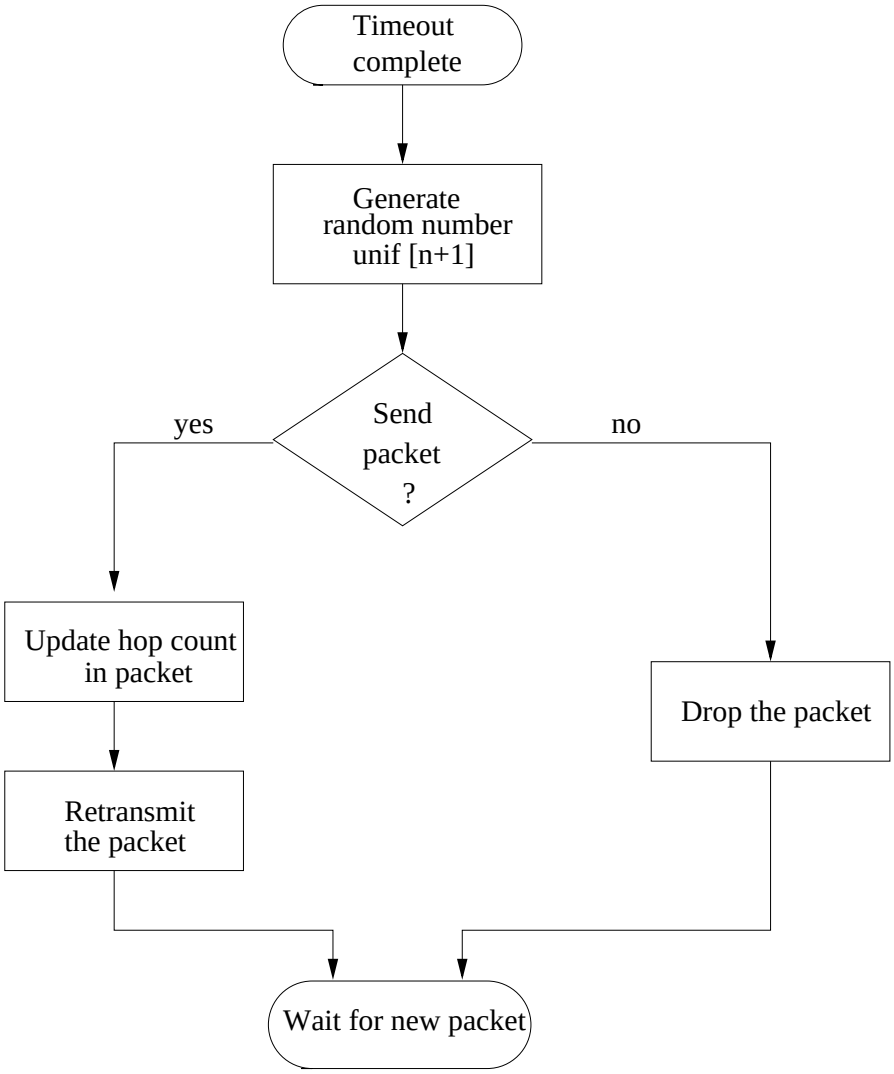

Figure 3.16: Procedure after completion of data wait period

3.11 Summary

In this chapter we have presented a framework for multicast routing in ad hoc networks using source grouped flooding. The protocol sets up source based flooding groups to the multicast group members. Nodes learn distance metrics during a request-response phase and use distance constraints to join the flooding group. Enhancements to optimize the size of the flooding group and a probabilistic data forwarding mechanism to reduce redundant data transmission have been proposed. Four algorithms have been described for evaluating the basic scheme and the enhancements proposed. The key features of the protocol are:

- Generation and maintenance of the flooding group based on distance constraints.
- A hop count based data forwarding scheme, which we expect to reduce overhead and improve channel access.
- A probabilistic data forwarding mechanism to reduce data redundancy and a method to obtain meaningful probabilities for this scheme adaptively from the network.
- Robustness to topology change due to the maintenance of multiple redundant paths.

Chapter 4

Performance Evaluation of Multicast Routing Protocols

4.1 Simulation Environment

OPNET 7.0 [17] discrete event engine was used to simulate our algorithms. The simulation modeled a network of nodes randomly placed within a $1000m$ x $1000m$ area. The network density i.e., the number of nodes in the network was varied as a simulation parameter. At the physical layer, radio propagation distance for each node was set to $250m$ and the channel capacity was $1Mbps$. Our model does not support radio capture [32] so, in the case of packet collisions all packets are dropped. The IEEE 802.11 Distributed Coordination Function (DCF) [16] as implemented in OPNET 7.0 was used as the Medium Access Control (MAC) protocol. The communication medium is broadcast and nodes have bi-directional connectivity. Each simulation was run for 100 seconds. Multiple runs were conducted with different seed values for each scenario and the collected data was averaged over

these runs. The multicast algorithms were developed as separate OPNET routing layer protocols.

4.1.1 Node Placement

Nodes in the network were randomly placed. Hence, network partitions can exist, irrespective of the denseness of the network. For a given experiment, all schemes are configured with the same seed value. Thus all schemes will encounter the same network scenario. Hence the performance of the schemes can be directly compared.

4.1.2 Mobility Model

All nodes in the network are mobile and move according to the "billiard mobility" model [33]. In this model nodes move at the set speed for a specified period of time towards a random destination, after this period a new random destination is chosen and the node moves towards this new destination with the same speed. If a node reaches the boundary of the fixed area, the node will rebound like a billiard ball and hence the name. In our simulations the boundary was set to the network dimension i.e. $1000m$ x $1000m$. Thus, the nodes are free to move to any region in the network area. The continuous movement of the nodes ensures regular change in the topology. This highly dynamic network is ideal to test the reliability of our algorithms.

4.1.3 Group Membership

Multicast group members are randomly chosen based on the seed value from the available set of nodes. Since our algorithms do not have explicit messages to join or leave the group, the members join the group at the beginning of the simulation. Sources are also randomly generated from the nodes in the network. The sources and group members are selected independently from the set of network nodes. Thus, there are no restrictions on the choice of sources and members. The number of sources and number of group members are simulation parameters. Experiments were conducted for a single multicast group.

4.1.4 Application Traffic

The OPNET 7.0 source generator model was used to generate Constant Bit Rate (CBR) traffic. The size of each data packet payload was 128 bytes. The number of such packets generated was varied as a simulation parameter. Source nodes start generating data at random instants of time, between 0 - DATA_START_TIME seconds. DATA_START_TIME is a configurable parameter and was set to 3 seconds. In our model, when a source initially wants to send data, it instructs the routing algorithm to establish the *flooding group*. The source then sends data packets at constant intervals of time. This interval is determined by the traffic load. In fact, the interval is the inverse of the load in *packets/sec*.

4.2 Simulation Methodology

4.2.1 Multicast Algorithms for Evaluation

Each simulation evaluated the following schemes:

4.2.1.1 Flooding

The basic flooding algorithm is chosen as a baseline algorithm for evaluating the performance of the source grouped flooding algorithms. In the *flooding* scheme, every node rebroadcasts each unique packet it receives. Multicast group members process the data packet. The *flooding* scheme is considered as a benchmark as it is shown to be robust and reliable against a wide range of mobility speeds.

4.2.1.2 Scheme Basic-SGFP

This is the Basic Source Grouped Flooding Protocol as described in Section 3.10.1. The scheme was implemented as described in the flowcharts. The scheme uses the basic *flooding group* creation method described in Section 3.1 and the hop count based data forwarding mechanism described in Section 3.5. This scheme is evaluated to understand the nature of source grouped flooding.

4.2.1.3 Scheme P-SGFP

This is the Probabilistic Source Grouped Flooding Protocol as described in Section 3.10.3. This scheme uses the probabilistic data forwarding mechanism de-

scribed in Section 3.7 along with the basic flooding group creation scheme. This scheme is evaluated to study the benefit and impact of the probabilistic forwarding scheme.

4.2.1.4 Scheme SP-SGFP

This is the Shortest Path Source Grouped Flooding Protocol as described in Section 3.10.2. This scheme creates the shortest path flooding groups as described in Section 3.6. This scheme is evaluated to study the impact of optimized flooding groups.

4.2.1.5 Scheme PSP-SGFP

This is the Probabilistic Shortest Path Source Grouped Flooding Protocol as described in Section 3.10.4. This scheme uses the probabilistic data forwarding mechanism described in Section 3.7 along with the shortest path flooding group creation scheme. This is the primary scheme being evaluated as it incorporates the basic approach to source grouped flooding and all the enhancements discussed in Chapter 3.

4.2.2 Simulation Attributes

The above schemes were evaluated as a function of the following attributes.

4.2.2.1 Protocol specific parameters

- Traffic load: The traffic is varied in terms of the number of data packets generated per second. The load is varied in terms of packets rather than packet size. The reason for this being, load variation in terms of packets would also alter the performance of the MAC layer. A node has to contend for the medium for every packet it wants to send.

- Refresh Interval: The route refresh interval is the frequency of generation of JOIN REQUEST messages. Each source generates a request message once every Refresh Interval seconds. The source grouped scheme does not require synchronized generation of request messages by all the sources. As the sources join the network at different times, they generate request messages at different times.

4.2.2.2 Multicast group parameters

- Number of members: The size of the multicast group or the number of members is varied.

- Number of sources: The number of sources in the network is varied.

4.2.2.3 Network parameters

- Mobility speed: The speed at which the network nodes move is varied. Variation against mobility determines the robustness of the protocols.

- Network density: The number of nodes in the network is varied while the network area is fixed at $1000m$ x $1000m$. The number of nodes is varied to study the impact of connectivity on the performance of the protocols.

4.2.3 Simulation Metrics

The collated data from the different runs were used to generate the following metrics. Some of these metrics were initially introduced in [20].

4.2.3.1 Goodput or Packet Delivery Ratio

The packet delivery ratio is defined as the number of data packets received by group members to the number of data packets supposed to be received by group members. This metric is an indication of the effectiveness of the protocol. Goodput and packet delivery ratio are used synonymously.

4.2.3.2 Data Overhead

The data overhead is measured in packets. It is defined as the ratio of the number of data packets transmitted in the network to the number of data packets received by the group members.

4.2.3.3 Control Overhead

The control overhead is defined in both packets and in bytes. This overhead is the ratio of the number of control packets transmitted to the number of data packets

received by the group members. This metric is also defined in bytes as the ratio of number of control bytes received to the number of data bytes received. Since MANETs are energy constrained networks [3, 22], it is important to characterize the overhead in terms of bits so as to quantify the energy used in transmitting control or redundant bits. Here we do not consider the bytes in the headers of data packets as control bytes. Hence control overhead defined in bytes is directly related to control overhead in packets.

4.2.3.4 Total Overhead

The total overhead is defined in packets. It is the sum of the data overhead and the control overhead in packets. This metric is an indication of the efficiency of the protocol.

4.2.3.5 Average End-to-End Delay

The end-to-end delay is the interval between the instant a source generates a packet and the time at which a member receives the packet. The end-to-end delay is aggregated for each packet for each member. The average per packet end-to-end delay is then calculated as the number of members and the number of packets received is known.

4.3 Simulation Results and Trade-off Analysis

In this section we present the simulation results for all the schemes as a function of the simulation attributes described in Section 4.2.2. The basic simulation model for all the experiments is as described in Section 4.1. Also we analyze the trade-off characteristics of these schemes.

4.3.1 Hop count based data forwarding

In this experiment, we evaluate the hop count based data forwarding mechanism as described in Section 3.5. We had hypothesized that, the hop count restricted forwarding would reduce the number of data retransmissions and possibly improve data delivery. We compare the performance of the hop count restricted data forwarding mechanism (HC restricted fwd) as described in Section 3.5, and the baseline data forwarding scheme (Normal fwd). In the baseline scheme, we do not compare the hop count in the data packet and the stored hop count in the node; all other aspects of the data forwarding mechanism is the same.

The network consisted of 50 nodes, randomly placed in an area $1000m$ x $1000m$. 20 members and 5 sources were randomly chosen from these 50 nodes. The refresh interval was set to $4secs$ and each source transmitted $2packets/sec$. Nodes in the simulation were mobile and moved at $5m/s$ as per the billiard mobility model.

Table 4.3.1 shows the goodput and the data overhead as defined in Section 4.2.3. Each of the four source grouped schemes as described in Section 4.2.1 were eval-

	Goodput		Data Overhead	
	Normal fwd	HC restricted fwd	Normal fwd	HC restricted fwd
basic-sgfp	.952	.925	2.235	2.146
p-sgfp	.914	.906	1.744	1.653
sp-sgfp	.92	.92	1.643	1.59
psp-sgfp	.90	.90	1.313	1.25

Table 4.1: Performance comparison of hop count restricted data forwarding and normal data forwarding

uated with the hop count restricted and the normal data forwarding schemes. From the table, it is clear that the hop count restricted forwarding scheme is more efficient, as expected. Scheme *basic-sgfp* has a considerable decrease in goodput, when hop count restricted forwarding is used. This is because, in *basic-sgfp* the size of the flooding group is large and therefore certain flooding nodes serve multiple group members. The redundancy in the normal data forwarding mechanism thus improves data delivery. The effect is less pronounced in *p-sgfp* which also creates large flooding groups but the redundancy is reduced by the probabilistic nature of data forwarding. Schemes *sp-sgfp* and *psp-sgfp* that create shortest path flooding groups per source seem to have the same goodput performance for both the hop count restricted data forwarding and normal data forwarding. Since, *psp-sgfp* is the primary scheme being evaluated, the hop count restricted data forwarding scheme is used as the base data forwarding scheme for all further experiments.

4.3.2 Mobility Speed

In this experiment, the mobility speed was varied between 0 to 30 m/s (0,5,10,20,30). The model consisted of 50 nodes randomly placed in the network. In each run of the simulation, all the schemes were tested on the same scenario, this was ensured by setting the same seed for all the scenarios. The seed was changed for different runs. 5 sources and 20 multicast group members were randomly chosen from these 50 nodes. The refresh interval for the source was set to 4 seconds. The source generated 2 packets/sec.

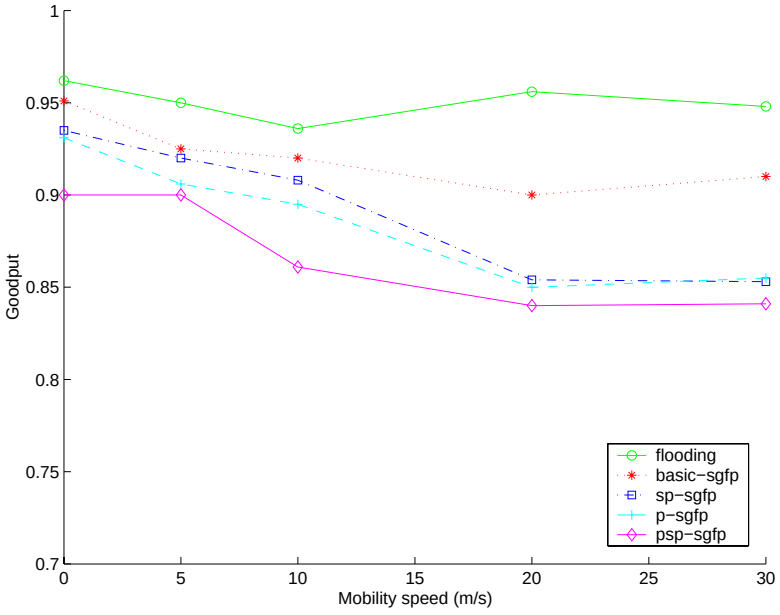

Figure 4.1: Packet Delivery Ratio Vs Mobility Speed

Figure 4.1 shows the packet delivery ratio as a function of the node mobility speed. It is seen that all the schemes show relatively stable performance under varying node speeds. The *flooding* scheme has the best goodput performance, hov-

ering around 96%. This high goodput value is due to the redundant transmission of data. The *basic-sgfp* scheme has performance close to that of *flooding* with goodput fluctuating around 94%. The reason being, the large size of the *flooding group* ensures reliability through redundancy. The size of the flooding group for this scheme is determined by the distance constraints 3.1 and (3.2) described in Section 3.1. The *sp-sgfp* scheme that creates shortest path flooding groups performs slightly worse than the *basic-sgfp* scheme. Since the flooding group here is generated based on distance constraint 3.3 described in Section 3.6, the routes are more susceptible to node movement. Thus the performance variation between these two schemes is more pronounced as the mobility speed increases. The probabilistic schemes *p-sgfp* and *psp-sgfp* recorded goodput values around 93% and 90% respectively. The increased packet loss in these schemes is due to unreliable nature of the probabilistic data forwarding mechanism. The goodput performance of these schemes seem to settle at 85% for networks with highly dynamic nodes. Thus the node mobility does not drastically affect the goodput performance of these schemes. An interesting point to note is the goodput values for zero mobility networks. Ideally *flooding* would be expected to record close to 100% goodput. In our experiments the value is 96% due to the presence of network partitions. Network partitions can drastically affect the performance of our schemes. As in the presence of partitions, redundancy does not imply reliability. Furthermore with the nodes being randomly placed, collisions at the MAC layer could result in network partitions on a per packet basis. This is particularly the case when a single node acts as a

bridge between two densely populated network regions displaced in space. Also node mobility being random could also result in network partitions. We expect our schemes to generate higher goodput values in fully connected networks i.e., networks in which a guaranteed path exists between two nodes, albeit via multiple hops.

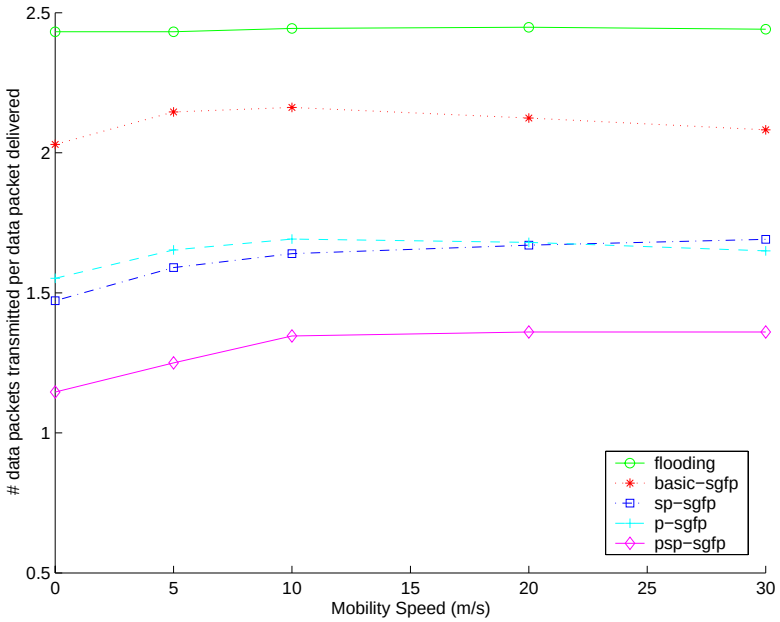

Figure 4.2: Data Overhead Vs Mobility Speed

Figure 4.2 shows the data overhead (in packets) variation against node mobility speed. The data overhead for all the schemes is almost constant against node mobility. The *flooding* scheme has the highest data overhead, 2.5 data packets transmitted for every data packet received. This is because, in flooding every node retransmits the data packet and therefore the redundancy is excessive. The *basic-sgfp* scheme has an overhead of 2 data packets for every data packets received.

This is due to the reduced size of the flooding group. Optimizing the size of the flooding in *sp-sgfp* reduces the number of redundant transmissions. The number of redundant transmissions is further reduced by the use of the probabilistic forwarding mechanism. Thus *p-sgfp* has data overhead of 1.5 compared to 2 of *basic-sgfp* and *psp-sgfp* has data overhead of 1.2 as compared to 1.5 of *sp-sgfp*.

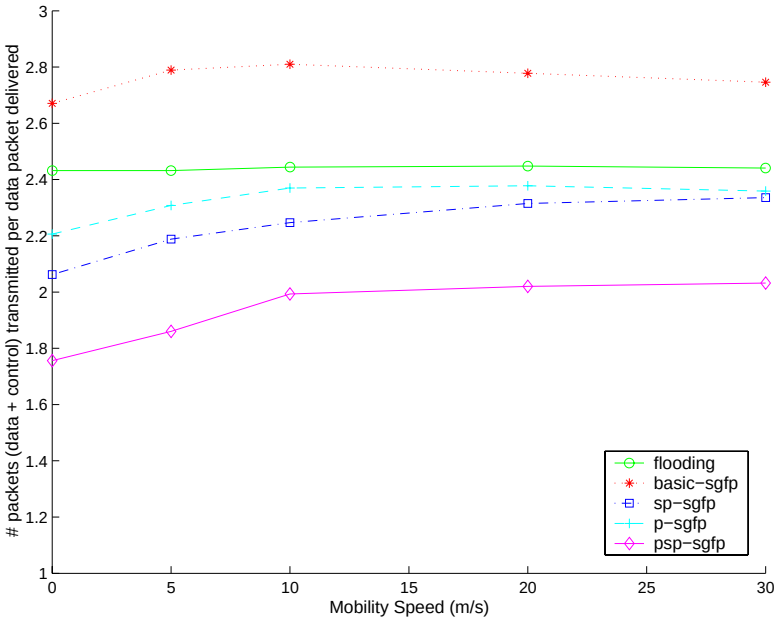

Figure 4.3: Total Overhead Vs Mobility Speed

Figure 4.3 shows the total overhead (data + control packets) variation against node mobility speed. The total overhead is an indication of the efficiency of the protocol. For the *flooding* scheme there is no control overhead so the total overhead is same as the data overhead. It is seen that the total overhead remains roughly the same for different node speeds. The total overhead of the *basic-sgfp* scheme is more than that of *flooding*. This is because, the size of the flooding group created

in this scheme results in considerable redundant transmission of data and also this scheme incurs control overhead during setup of the flooding group. The other source grouped flooding schemes are more efficient than flooding. Particularly the *psp-sgfp* scheme has a considerably lower overhead, around 1.9 packets transmitted as opposed to 2.5 of *flooding*.

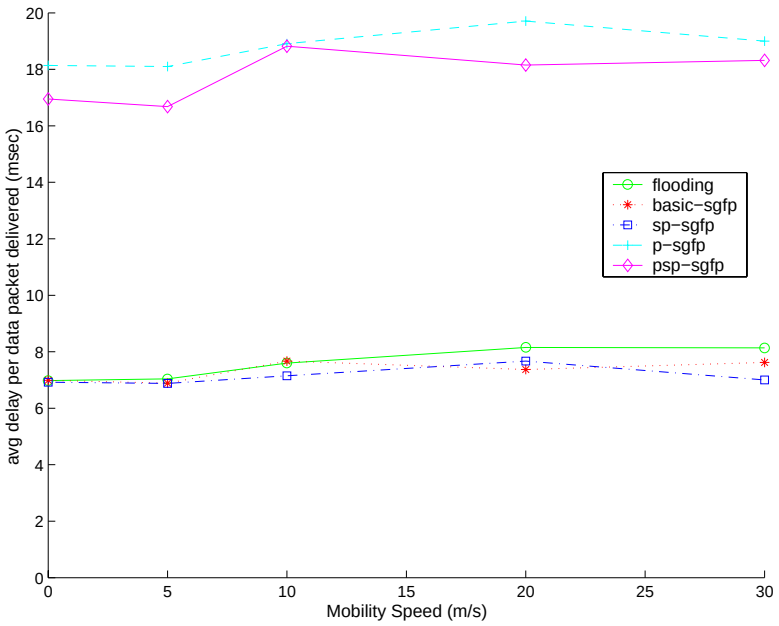

Figure 4.4: Average End-to-End Delay Vs Mobility Speed

Figure 4.4 shows the end to end delay in receiving a packet as a function of node speed. The end to end delay is an important factor in real time applications like voice and video. Typically delays of $100msec$ are acceptable for such applications [34]. It is seen that schemes *flooding*, *basic-sgfp* and *sp-sgfp* have an average per packet end to end delay of $7msec$. The delay hovers around this value as node speed increases. The end to end delay captures the delay involved in acquiring the

medium, propagation delay and buffering delay for each hop. The probabilistic schemes *p-sgfp* and *psp-sgfp* have higher end to end delay of around 17$msec$. This increased in delay is due to the probabilistic nature of data forwarding, wherein nodes wait for a random period of time for duplicates before deciding to transmit the packet. The contribution of the wait period on a per hop basis results in higher end-to end delay.

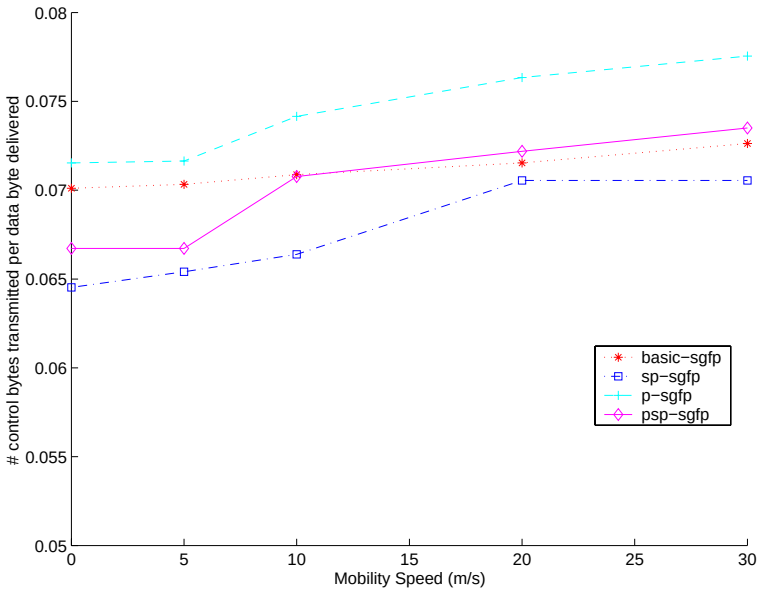

Figure 4.5: Control Overhead Vs Mobility Speed

Figure 4.5 shows the control overhead in bytes. This measure gives an indication of energy usage to setup the flooding group. It is seen that all the source grouped flooding schemes roughly introduce the same control overhead in bytes per data byte delivered. This value is roughly around 0.07 bytes per data byte delivered. Thus the energy expended in sending out control information is neg-

ligible compared to the data received. This measure depends on the number of control packets (JOIN REQUEST and JOIN REPLY messages) transmitted. This is influenced by collisions in the network and network partitions. It is seen that in the probabilistic schemes, more control bytes are transmitted per data byte delivered. This indicates that the probabilistic schemes alleviate MAC contention and collision to some extent as hypothesized in Section 3.7. Also it is seen that the number of control bytes transmitted in the schemes increases with mobility. This is probably due to the reduced number of data bytes received as mobility increases.

4.3.3 Number of Sources

Here, we present the results for the various simulation metrics as a function of the number of sources. The number of sources was varied from 1 to 20 (1,2,5,10,20). The case of 1 source corresponds to one-many communication (eg., classroom lecture) and the case of 20 sources can be that of a conference meeting. The network comprised of 50 nodes and 20 of these were randomly chosen as multicast group members. The mobility speed was set to $5m/s$ for each node. The refresh interval for the source was set to 4 seconds and each source generated $2packets/sec$. As before, for each run all the schemes the same scenario. This experiment was designed to understand the scalability of the schemes. Increasing the number of sources increases the data and control traffic as well as the number of flooding groups created.

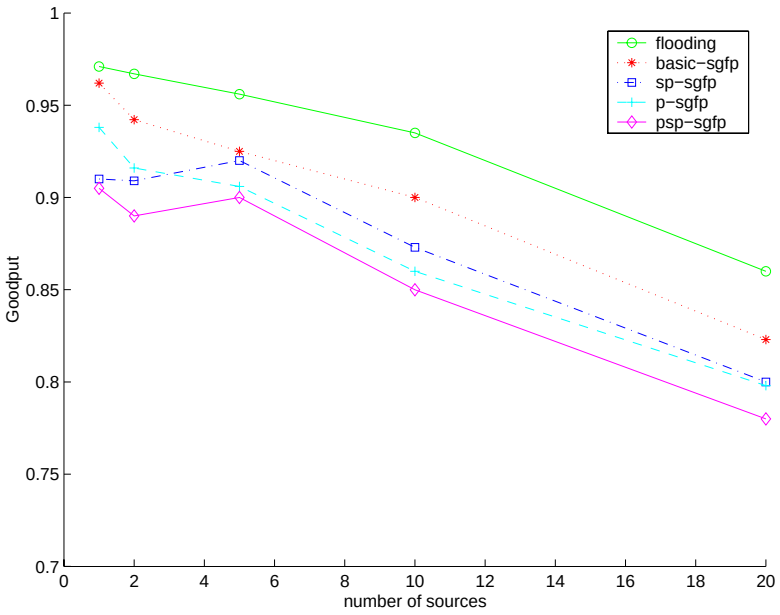

Figure 4.6: Goodput Vs Number of Sources

Figure 4.6 shows the variation of Goodput as a function of the number of sources sending data to the multicast group. As expected, the packet delivery ratio decreases as the number of sources increase. The reasons being, increased data traffic and increased control traffic. This increase in traffic results in increased collisions which in turn affects the route generation process. Interestingly the degradation in performance is linear and gradual with a slope of around -0.007 for all the schemes. Thus the schemes should scale with increase in number of sources, with an acceptable deterioration in goodput performance. Again, best goodput achievable is limited due to the presence of network partitions. The schemes *basic-sgfp* and *p–sgfp* that use the basic source grouped flooding scheme to create the groups exhibit a similar curve. The same is true for *sp-sgfp* and

psp-sgfp as well, these schemes create the shortest path groups. The probabilistic schemes perform slightly worse than their non-probabilistic counter-parts. This is due to the unreliable nature of the probabilistic forwarding scheme.

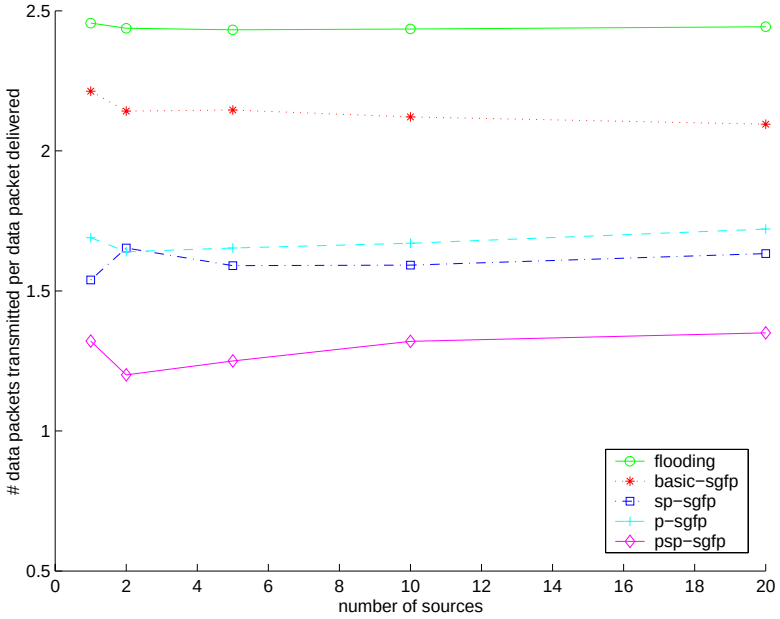

Figure 4.7: Data Overhead Vs Number of Sources

Figure 4.7 shows the variation of the data overhead against the number of sources. The number of data packets generated in the network increases with the number of sources. From the graph it is clear that the data overhead as defined in Section 4.2.3.2, remains almost constant as the number of sources increase for all the schemes. This indicates that the success rate of data packets decreases as the number of sources increase. This is probably because of the collisions in the network resulting in reduced rate of retransmissions in the network. The data overhead thus remains the same for each packet delivered irrespective of the number

of sources.

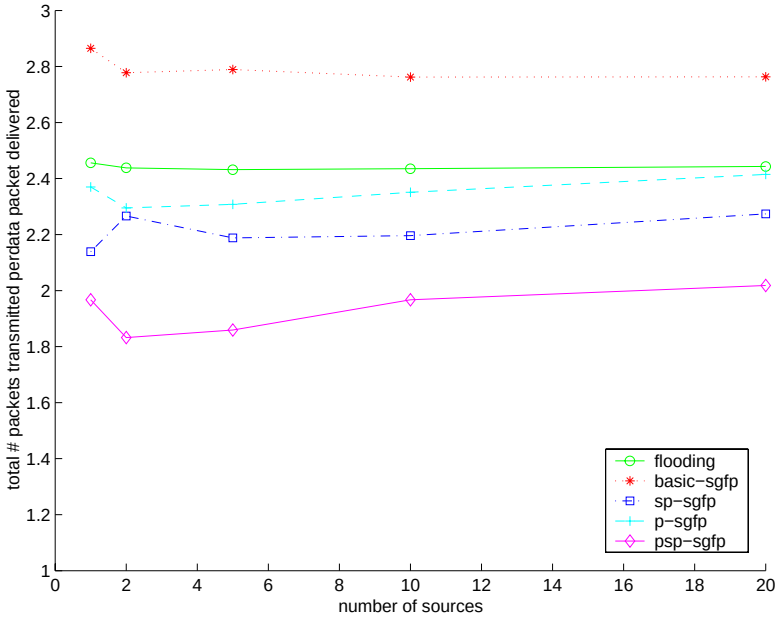

Figure 4.8: Total Overhead vs Number of Sources

Figure 4.8 shows the variation of the total overhead (data + control) as the number of sources increase. The total overhead remains constant for all the schemes for different number of sources in the network. The number of control packets generated varies linearly with the number of sources. Like the data overhead the control overhead also remains constant when collected relative to the number of data packets delivered. This again is due to increased collisions in the network. The reduced rate of control and data packets retransmitted due to the collisions results in reduced packet delivery to the members. Hence the total overhead as defined in Section 4.2.3.4, remains constant as the number of sources increase.

Figure 4.9 shows the average end to end delay for packet delivered to a mem-

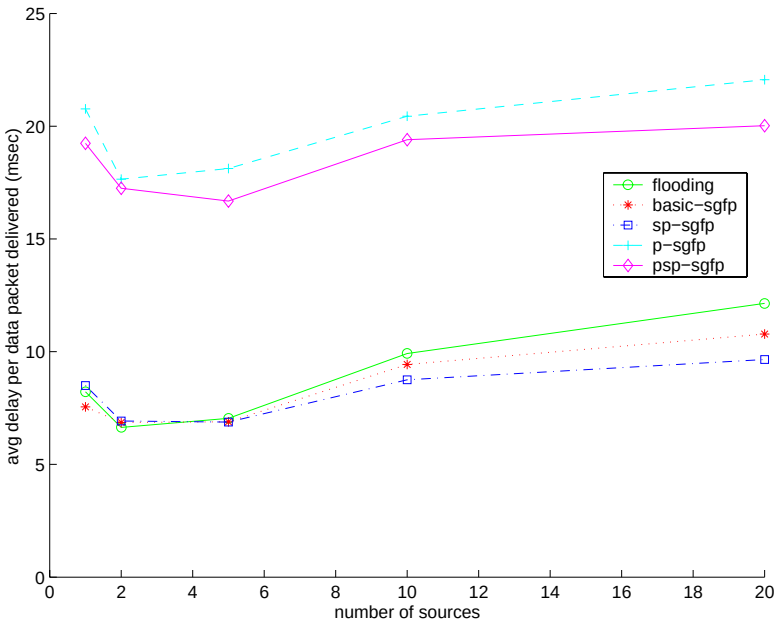

Figure 4.9: Average End-to-End Delay Vs Number of Sources

ber. The average end to end delay for the non-probabilistic schemes is roughly around $7msec$ while the average end to end delay for the probabilistic schemes which use the data waiting mechanism is around $20msec$. The delay for all the schemes increase slightly with increase in number of sources. This is because of the increased medium access time. Thus with increased traffic, the MAC layer not only experiences increased collisions but also increased delays.

Figure 4.10 shows the control overhead in bytes. The control overhead is negligible for all the schemes and remains fairly constant for varying number of sources in the network. The overhead being roughly 0.07 bytes per data byte delivered. Hence the energy expended in sending control information is negligible.

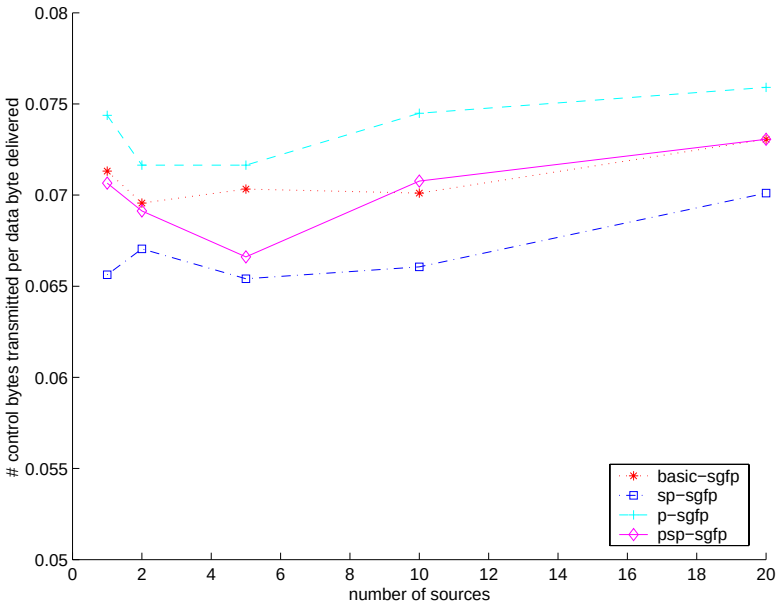

Figure 4.10: Control Overhead Vs Number of Sources

4.3.4 Route Refresh Interval

In this experiment the refresh interval is varied between 3 seconds to 10 seconds (3,4,6,8,10). Each source sends out a JOIN REQUEST message once at the beginning of every refresh interval. The network comprised of 50 nodes randomly placed. 5 of these 50 were randomly chosen as sources and another 20 were chosen as group members. The nodes moved at a speed of $5m/s$. Each source generated $2packets/sec$. Since the *flooding* scheme does not generate any control information, it is not considered for performance comparisons.

Figure 4.11 shows the goodput variation as a function of the source route refresh interval. The goodput remains almost the same for all the schemes as the refresh interval is increased. This indicates that the *flooding group* mesh structure is stable

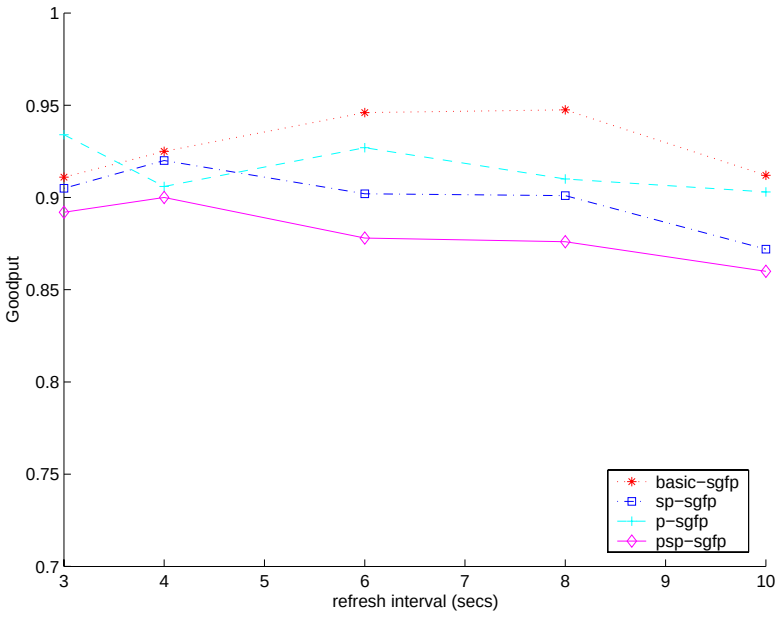

Figure 4.11: Goodput Vs Refresh Interval

and does not have to be refreshed regularly. In fact with increased refresh interval, the goodput for schemes *basic-sgfp* and *p-sgfp* increases marginally. This is because, the reduced frequency of control message generation reduces MAC contention and collision and therefore increases the packet delivery rate. In the case of *sp-sgfp* and*psp-sgfp* which generate shortest path flooding groups, the refresh interval does not affect the goodput. This is probably due to the even balance between the MAC layer benefits and the performance degradation due to out dated routes in the flooding groups. For a small refresh interval ($3secs$) the probabilistic schemes fare better or as good as the non-probabilistic schemes. The non-probabilistic schemes would incur losses due to MAC contention and collisions due to frequent transmission of control packets. Where as, the probabilistic schemes, alleviate

to some extent the medium access problems resulting in improved performance; though these schemes are less reliable.

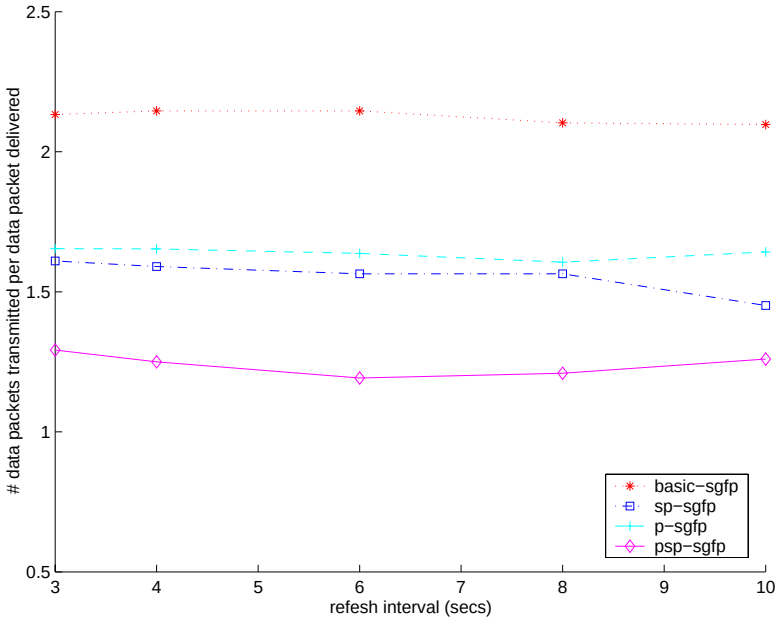

Figure 4.12: Data Overhead Vs Refresh Interval

Figure 4.12 shows the data overhead variation against refresh interval. The data overhead for all the schemes is almost constant as the refresh interval is varied. This is due to stability of the flooding groups generated by these schemes.

Figure 4.13 shows the total overhead as a function of the refresh interval. The total overhead for all the schemes, reduces as the refresh interval is increased. This is expected as the number of control packets decreases with increased refresh interval. The total overhead for scheme psp-sgfp is as low as 1.6 when the refresh interval is set to 10*secs* i.e; when JOIN REQUEST messages are sent every 10*secs*.

Figure 4.14 shows the control overhead in bytes for different value of the refresh

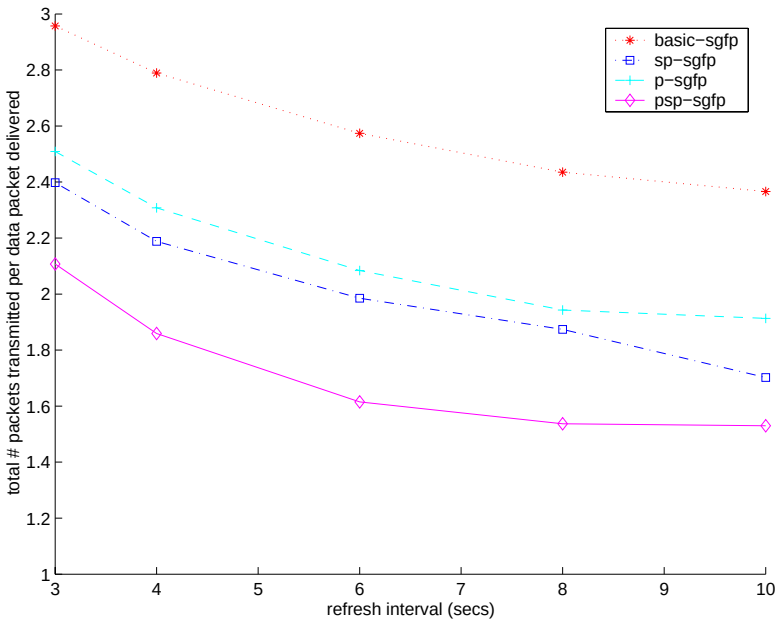

Figure 4.13: Total Overhead Vs Refresh Interval

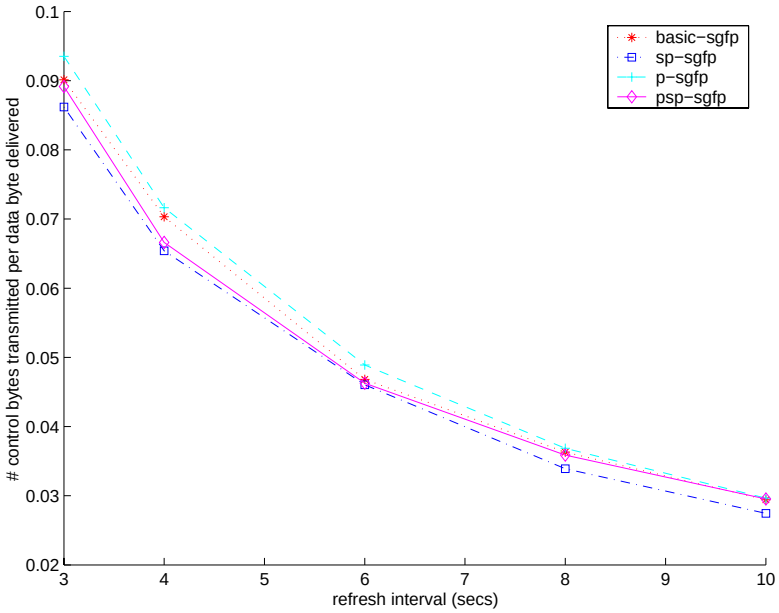

Figure 4.14: Control Overhead Vs Refresh Interval

interval. As expected for all the schemes,the number of control bytes transmitted reduces sharply as the refresh interval is increased.

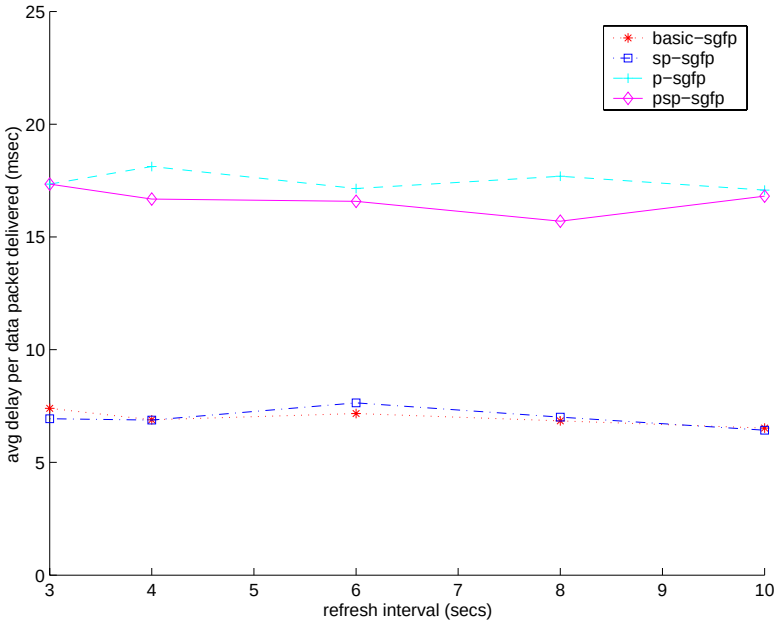

Figure 4.15: Average End-To-End Delay Vs Refresh Interval

Figure 4.15 shows the end-to-end delay as a function of the refresh interval. The average delay is constant for the different schemes. The probabilistic schemes record higher delays due to the wait interval involved in the probabilistic forwarding mechanism.

4.3.5 Multicast Membership Size

In this experiment the number of multicast group members was varied between 10 and 40 (10,15,20,30,40). The network comprised of 50 nodes placed randomly. 5 of these 50 were randomly chosen to be source nodes. The members were also

randomly chosen from the 50 nodes. Mobility speed for each node was to $5m/s$. Each source generated $2packets/sec$. The experiment was designed to study the performance of the schemes as a function of the group membership.

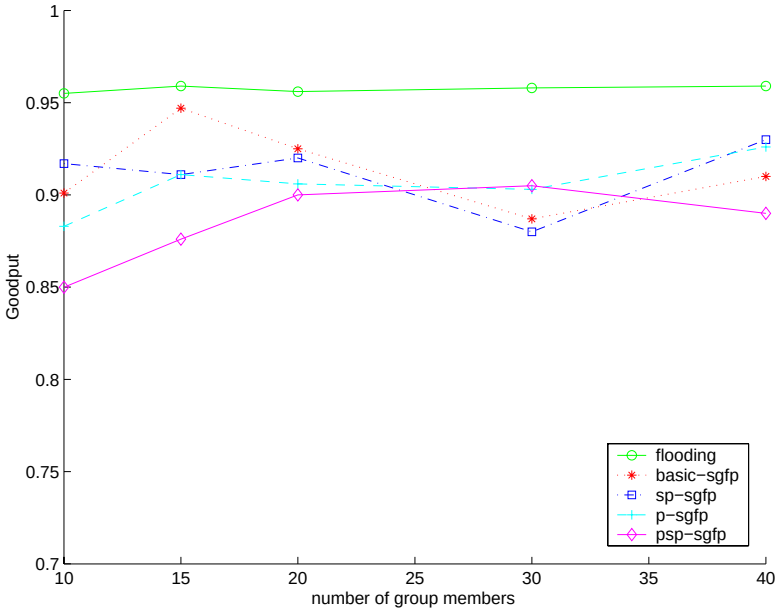

Figure 4.16: Goodput Vs Multicast Group Size

Figure 4.16 shows the goodput variation as a function of the group membership. The *flooding scheme* shows a steady performance as the number of members is increased. The reason being, in *flooding*, every node forwards the packet therefore irrespective of whether a node is a member or not, it receives the data packet. The goodput for the source grouped schemes, fluctuates between 0 - 10% of the goodput of *flooding*. Scheme *basic-sgfp* has a higher goodput when the group membership is small (10 - 20). The performance deteriorates as the number of members increases. This is probably due to the increased generation of control messages (JOIN RE-

PLY messages). The resulting increase in the size of the flooding group results in increased MAC contention and collisions. Scheme *psp-sgfp* on the other hand, displays improved goodput performance as the group membership size increases. The goodput steadies at 90% for group sizes of 20, 30 and 40. Scheme *p-sgfp* has almost constant goodput. The probabilistic schemes have a higher goodput performance as the group size increases. The is probably due to a reduction in redundant data transmission and improved MAC performance. Scheme *sp-sgfp* has a goodput performance curve similar to that of *basic-sgfp*. As *sp-sgfp* creates shortest path flooding groups, the reduced size of the flooding group improves goodput for large multicast groups.

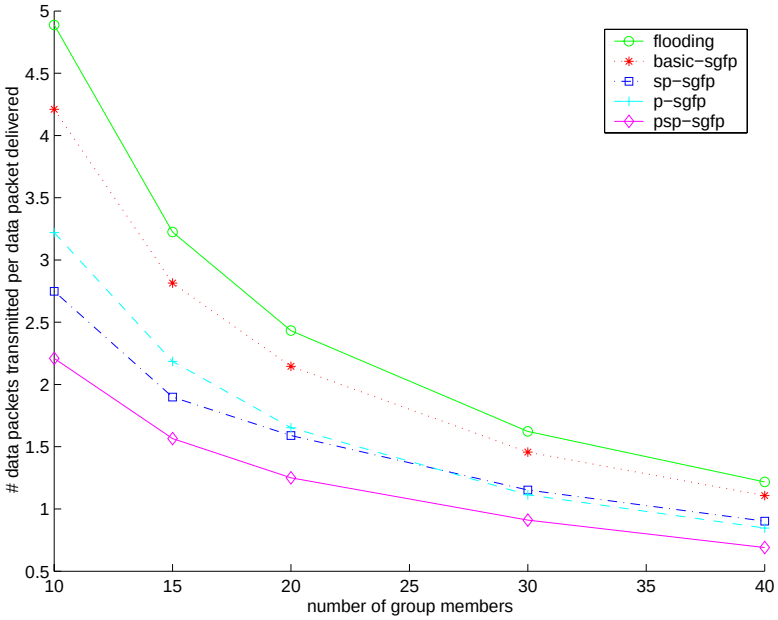

Figure 4.17: Data Overhead Vs Multicast Group Size

Figure 4.17 shows the data overhead variation as the multicast group size in-

creases. The data overhead for all the schemes decrease rapidly as group membership size increases. The reason being, as the group size increases more nodes are valid multicast receivers, thus increasing the number of data packets delivered. Also there is considerable overlap of flooding nodes and group members. Thus each retransmitted data packet has a high probability of reaching a group member and hence is not a redundant transmission. The data overhead for the different schemes seems to converge as the group size increases. The source grouped schemes have a lower overhead than *flooding*. Scheme *psp-sgfp* has an appreciable improvement over *flooding*, even when the group size is 40.

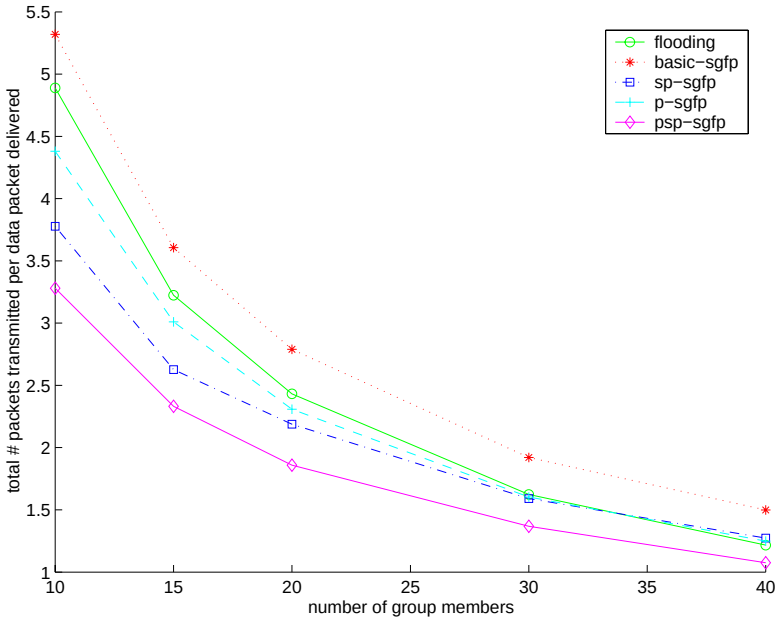

Figure 4.18: Total Overhead Vs Multicast Group Size

Figure 4.18 shows the total overhead variation as a function of the group size. The *flooding* scheme has no control overhead. Thus the total overhead for *flooding*

is same as the data overhead. For the source grouped schemes as the group membership size increases, the number of reply messages generated increases. The total overhead for all the schemes, decreases as the group size increases. This is because the number of data packets delivered to the members increases, as the group size increases. Scheme *psp-sgfp* has a better performance compared to *flooding*. This is due to the reduced number of retransmissions as a result of the shortest path flooding group and probabilistic forwarding. The other source grouped schemes fare worse or almost the same as *flooding*.

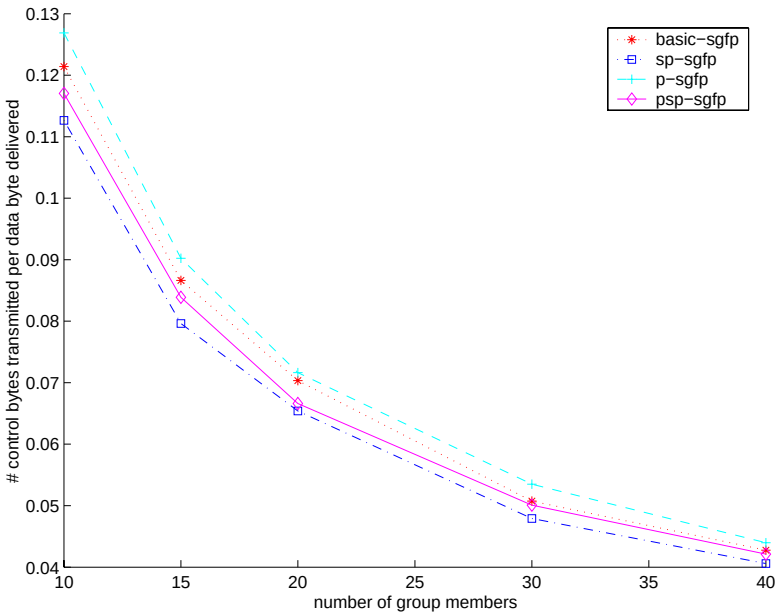

Figure 4.19: Control Overhead Vs Multicast Group Size

Figure 4.19 shows the control overhead as a function of the group size. The control overhead decreases as the group size increases. This is due to the increased number of data packets delivered to members. Thus as the number of members

increases, the control overhead relative to per data packet delivered reduces. The probabilistic schemes *p-sgfp* and *psp-sgfp* due to reduced retransmission of redundant data packets alleviate MAC problems. This improves the success rate of control packets. Thus the probabilistic schemes have more control overhead than the non-probabilistic schemes.

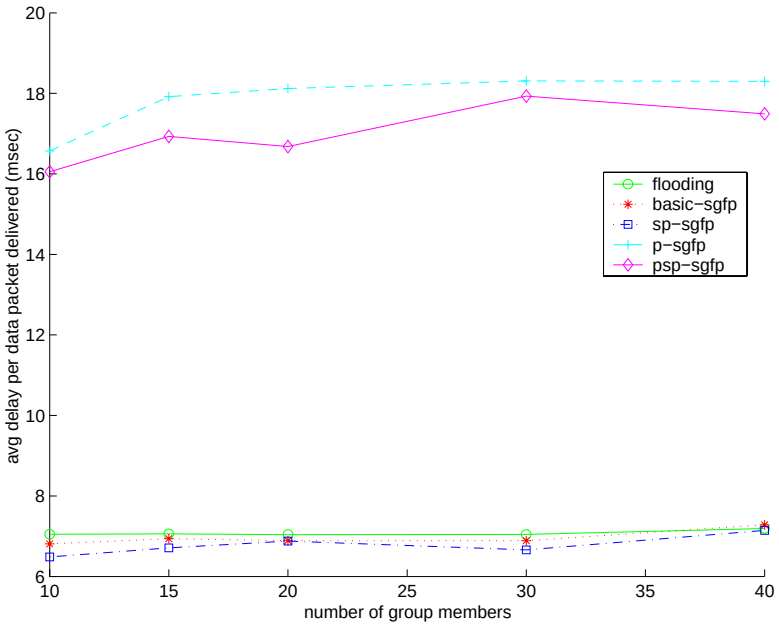

Figure 4.20: Average End-To-End Delay Vs Multicast Group Size

Figure 4.20 shows the end-to-end delay as the group membership increases. The delay remains almost constant for all the schemes. Schemes *basic-sgfp* and *sp-sgfp* have average end-to-end delay, same as that for *flooding*. The probabilistic schemes have increased delay due to the data wait timer involved in the probabilistic data forwarding mechanism.

4.3.6 Traffic Load

In this experiment, the traffic generated by each source was varied from 2 to 10 $packets/sec$ (2,4,6,8,10). The network comprised of 50 nodes randomly placed in the network. 5 of these 50 were randomly chosen as sources and 20 were chosen as multicast group members. Each node in the network moved at $5m/s$. The source refresh interval for generating route refresh packets was set to $4secs$.

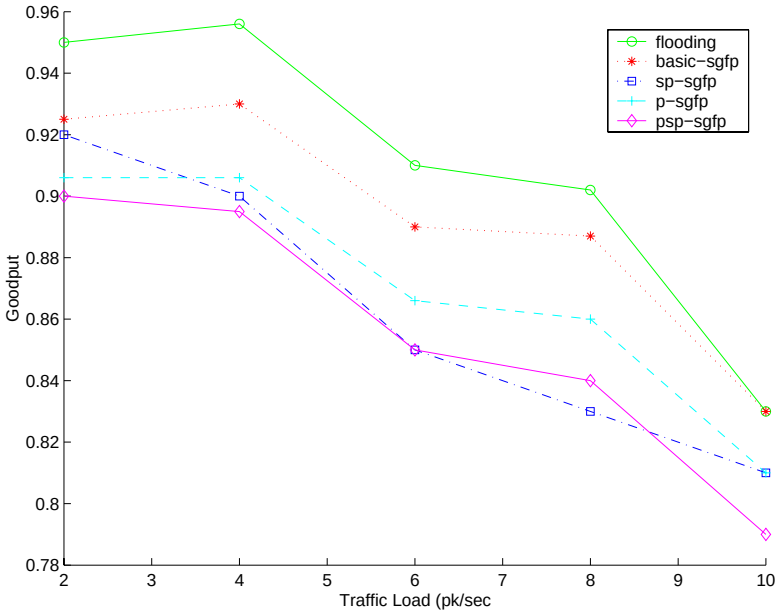

Figure 4.21: Goodput Vs Traffic Load

Figure 4.21 shows the variation of the goodput under varying traffic load conditions. The goodput deteriorates steadily for all the schemes. This is due to the increased number of collisions resulting from increased and more regular contention for the medium. When the load is $10packets/sec$, each source sends out a packet every $0.1seconds$. This implies that the network is very active almost all the time,

i.e., every flooding node is always contending for the medium. This excessive contention results in loss of data and control packets. The reduced success rate of control packets results in out dated routes in the flooding group; resulting in reduced data delivery. At $2 packets/sec$ the probabilistic schemes *p-sgfp* and *sp-sgfp* perform slightly worse than the other schemes due to reduced retransmission of data packets.

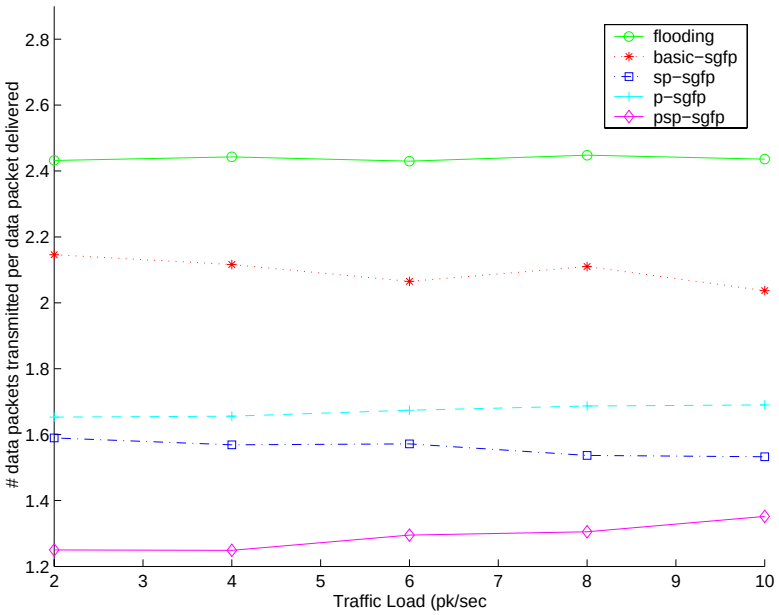

Figure 4.22: Data Overhead Vs Traffic Load

Figure 4.22 shows the data overhead as a function of the traffic load. The data overhead is almost constant for all the schemes. Increased traffic results in more number of data packets transmitted and more data packets delivered to the members. As the load increases, packet loss due to collisions and contention increases. The data overhead ratio remains constant as the loss of data packets

results in corresponding loss in data delivered.

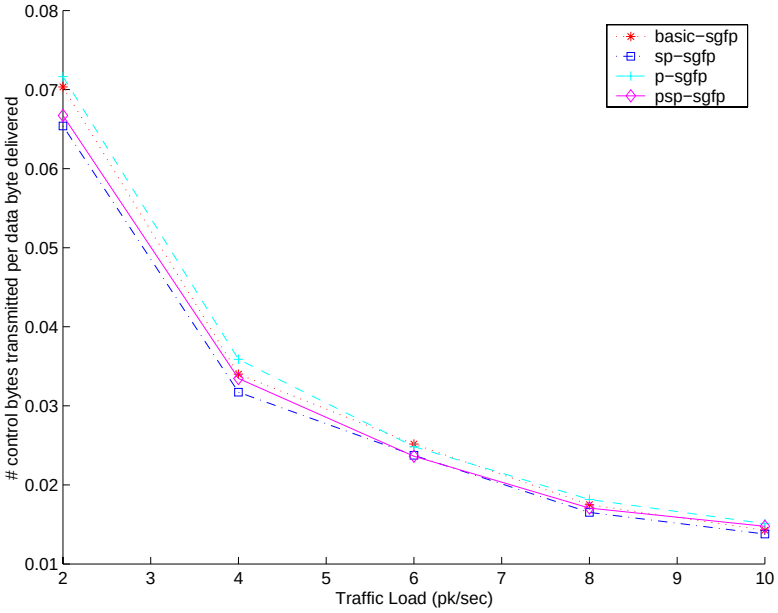

Figure 4.23: Control Overhead Vs Traffic Load

Figure 4.23 shows the variation of the control overhead (in bytes) as a function of the traffic load. The control bytes transmitted decreases with the increase in traffic load. This is because, the number of control packets generated does not depend on the traffic load. Thus the with more data packets delivered to the members, the per packet and per byte control overhead decreases. Also, the number of control packets transmitted may be reduced due to increased channel access and usage as the traffic increases.

Figure 4.24 shows the total overhead variation as a function of the traffic load ($packets/sec$). The total overhead for *flooding* remains constant as in this scheme no control information is generated. For the source grouped flooding schemes the

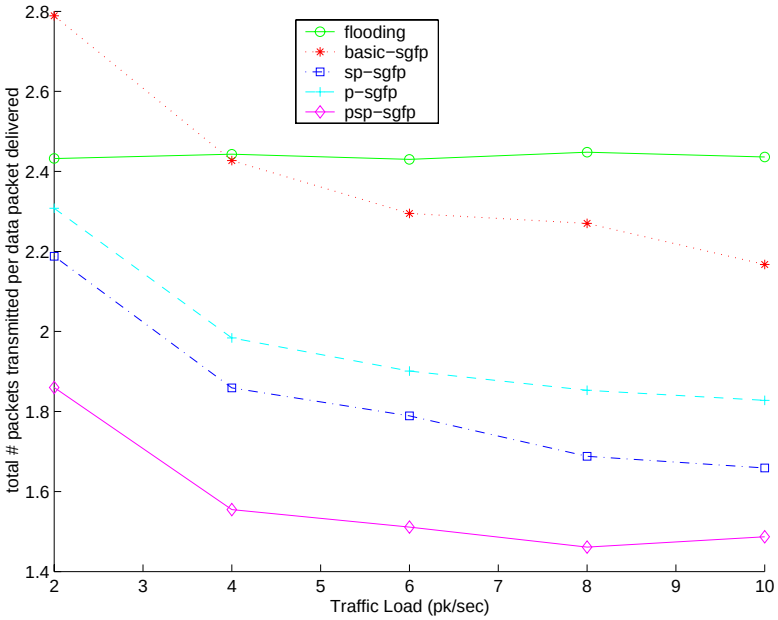

Figure 4.24: Total Overhead Vs Traffic Load

total overhead decreases with increased traffic load. We saw in figure 4.22 that the data overhead is almost constant. Therefore the decrease in the total overhead is due to the reduced number of control packets transmitted in the network. This also explains the reduced goodput performance of the schemes.

Figure 4.25 shows the end-to-end delay as the traffic load increases. The delay increases for all the schemes. This is due to the increased delay in accessing the medium and buffering delay on a per hop basis.

4.3.7 Network Density

In this experiment the node density is varied i.e, the number of nodes in the network with area fixed at $1000m$ x $1000m$. The node density was varied between 30 to

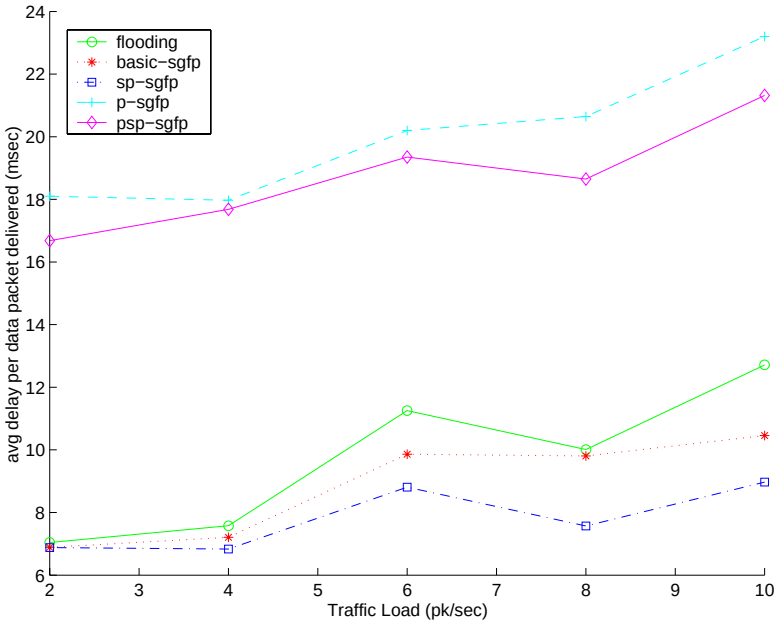

Figure 4.25: Average End-To-End Delay Vs Traffic Load

70 nodes (30,40,50,60,70). This experiment was designed to study the impact of network partitions and connectivity in the network. 5 source were randomly chosen as sources and 20 nodes were chosen as multicast group members. The mobility speed for the nodes was set to $5m/s$. Each source generated $2packets/sec$. Since the node placements in the network is random, it is possible that in certain scenarios, partitions are more prominent than in others. However, for a particular scenario, all schemes encounter the same connectivity pattern.

Figure 4.26 shows the goodput variation for networks with different node densities. We see that the goodput performance for all schemes, improves rapidly as the node density increases. For a network with 30 nodes we see that the goodput for *flooding* is around 70%. The goodput of the source grouped flooding schemes

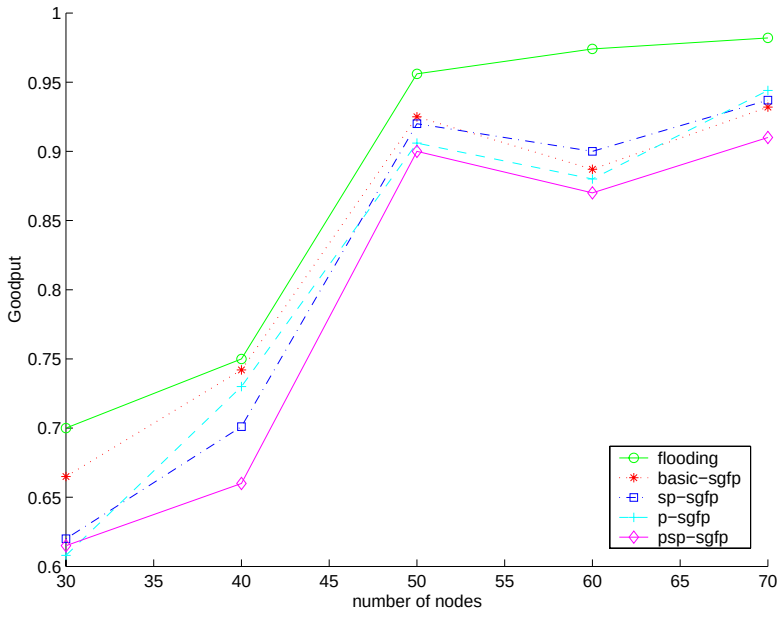

Figure 4.26: Goodput Vs Network Density

is between 60% to 65%. This implies substantial partitions in the network. At 40 nodes there is a 5% improvement in performance for all the schemes. The goodput improves significantly for node densities of 50, 60 and 70 nodes. The number of collisions and MAC contention increases with the number of nodes. The *flooding* algorithm being highly redundant is able to record a better performance as long as connectivity is improved. The source grouped flooding schemes are affected by the collisions and hence goodput seems saturated between 90% to 95%. The goodput of the source grouped schemes is within 10% of that of *flooding*. As the number of nodes increases there is hardly any difference between the goodput performance of the different source grouped schemes. From the figure it is clear that the increased number of nodes improves the connectivity of the network. However, node place-

ments in the network is random and since we have considered only 3 samples of the network, these results cannot be generalized.

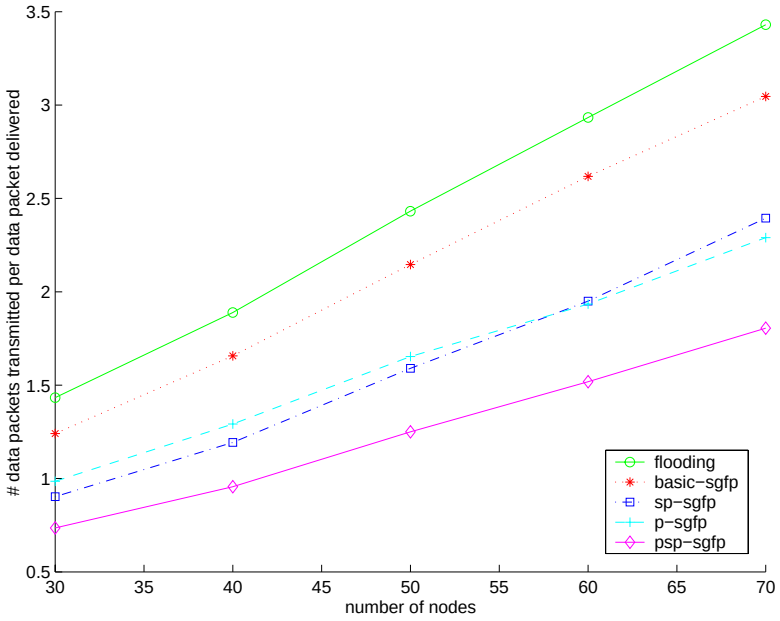

Figure 4.27: Data Overhead Vs Network Density

Figure 4.27 shows the data overhead as a function of node density. The data overhead for all the schemes, increases steadily as the number of nodes in the network increases. This is expected as the size of the flooding group increases with increase in the number of nodes. The slope of *p-sgfp*, *sp-sgfp* and *psp-sgfp* is less than that of *flooding* and *basic-sgfp*. This is due to the reduced size of the flooding group in the shortest path algorithms. The probabilistic mechanism of data forwarding also reduces redundant data retransmission. The data overhead in the case of 70 nodes, is least for *psp-sgfp*, only 1.5 compared to 3.5 in *flooding*.

Figure 4.28 shows the total overhead variation as a function of number of nodes

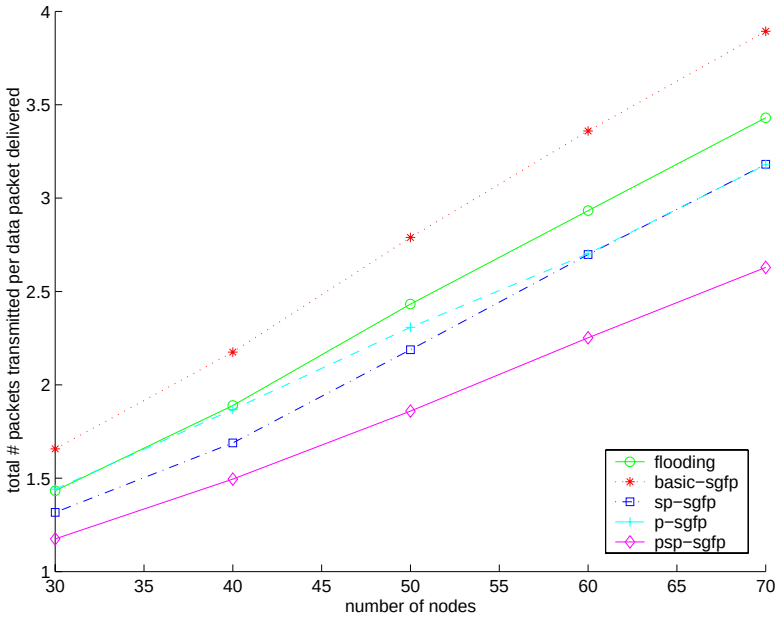

Figure 4.28: Total Overhead Vs Network Density

in the network. The total overhead increases for all the schemes as the number of nodes in the network increases. The *flooding* scheme has no control overhead. Scheme *basic-sgfp* has a higher total overhead than *flooding* due to the large size of the flooding group resulting in more data retransmissions and also the control packets exchanged to create and maintain the group. Schemes *p-sgfp*, *sp-sgfp* and *psp-sgfp* have lesser overhead than *flooding*. The difference is more pronounced as the number of nodes is increased. Scheme *psp-sgfp* has the least total overhead due to the reduced size of the flooding group and due to the probabilistic data forwarding mechanism.

Figure 4.29 shows the control overhead as a function of the number of nodes. The control overhead increases for all the schemes with increase in number of nodes.

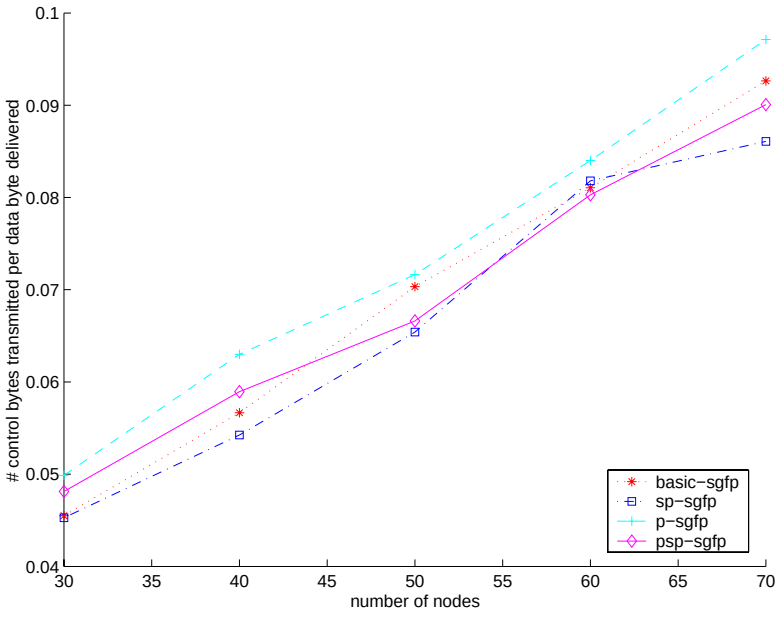

Figure 4.29: Control Overhead Vs Network Density

This is expected as the control messages are broadcasted in the network. We see that the probabilistic schemes *p-sgfp* and *psp-sgfp* have transmitted more control packets than their non-probabilistic counterparts. This is due to the reduced MAC contention and collision achieved due to reduced retransmissions in these schemes.

Figure 4.30 shows the end-to-end delay as the number of nodes in the network increases. The delay for all the schemes is relatively constant. The probabilistic schemes have a higher delay due to the data wait interval involved in probabilistic forwarding. The other source based schemes have delay characteristics similar to that of *flooding*.

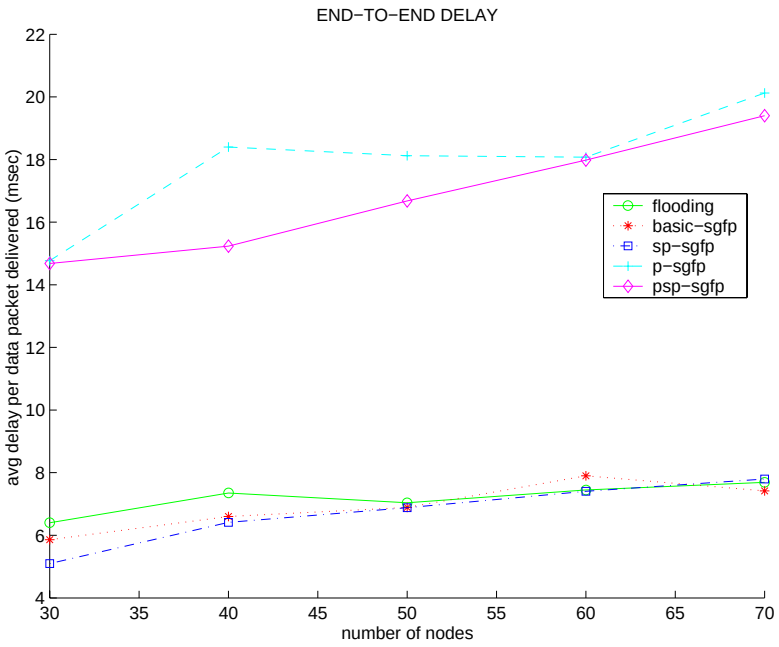

Figure 4.30: Average End-To-End Delay Vs Network Density

4.3.8 Trade-offs in Performance

In this section we analyze the trade-off between the effectiveness (goodput) and the efficiency (total overhead) of the protocols. From the results, we can suggest suitable operating values for certain characteristics or parameters of the MANET. The data represented in this section is derived from the results presented in the previous sections. Thus, the simulation setup for each of the trade-offs curves is same as defined in the previous sections for the corresponding simulation attribute.

Figure 4.31 shows the trade-off between the goodput and total overhead for different mobility speeds. The *flooding* scheme has best goodput performance for all mobility speeds. The total overhead remains unchanged as a function of

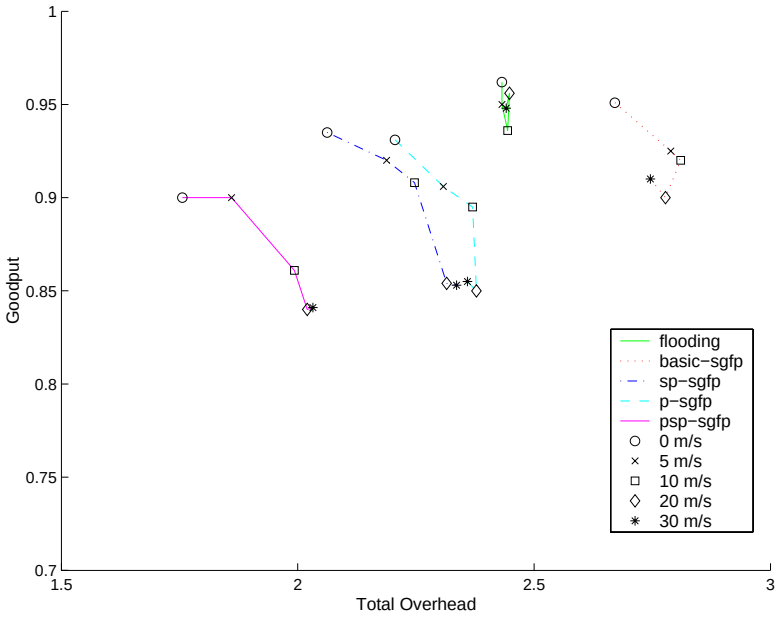

Figure 4.31: Trade-off curve for different mobility speeds

mobility. The mobility range 0-5m/s seems the ideal range to operate the source grouped protocols. In this range, the protocols are most effective (best goodput) and most efficient (least total overhead). We can see that scheme *psp-sgfp* is the most efficient protocol while still providing goodput within 10% of *flooding*.

Figure 4.32 shows the trade-off between goodput and total overhead for different number of sources generating packets to the multicast group. We can see that the total overhead for all the schemes remains almost the same. The goodput decreases linearly with increase in the number of sources. All schemes show small variations in goodput for 1-5 sources in the network. The goodput decreases slightly for 10 sources. Thus the effectiveness of the protocols is relatively stable for 1-10 sources. Scheme *psp-sgfp* is the most efficient scheme. It also achieves a goodput within 6%

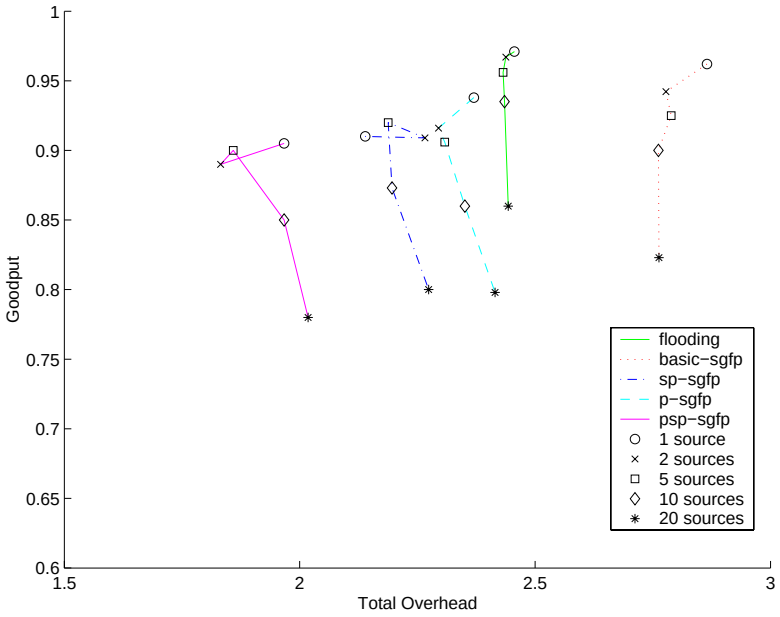

Figure 4.32: Trade-off curve for different number of sources

of that of *flooding*

Figure 4.33 shows the trade-off between goodput and total overhead for different multicast group sizes. The *flooding* scheme records a constant goodput as the group size increases. When there are 40 group members, the total overhead is around 1.2, indicating that flooding is efficient when 70% or more, of the nodes in the network are group members. The source grouped flooding schemes all have goodput within 6-10% of *flooding*. However, only *psp-sgfp* is consistently more efficient than *flooding*. All these schemes seem effective for MANETs where more than 40% of the nodes are group members, with *psp-sgfp* being the most efficient.

Figure 4.34 shows the trade-off between goodput and total overhead for different values of the route refresh interval. The *flooding* scheme is not considered for this

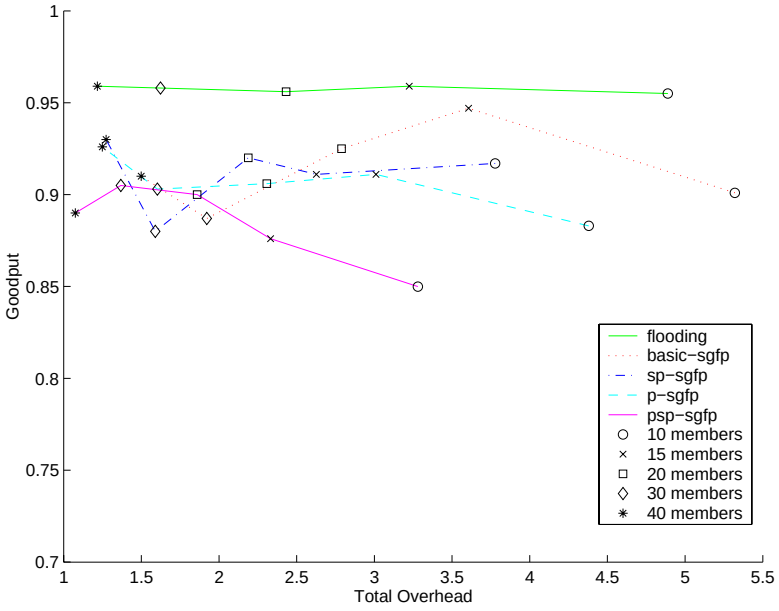

Figure 4.33: Trade-off curve for different number of members

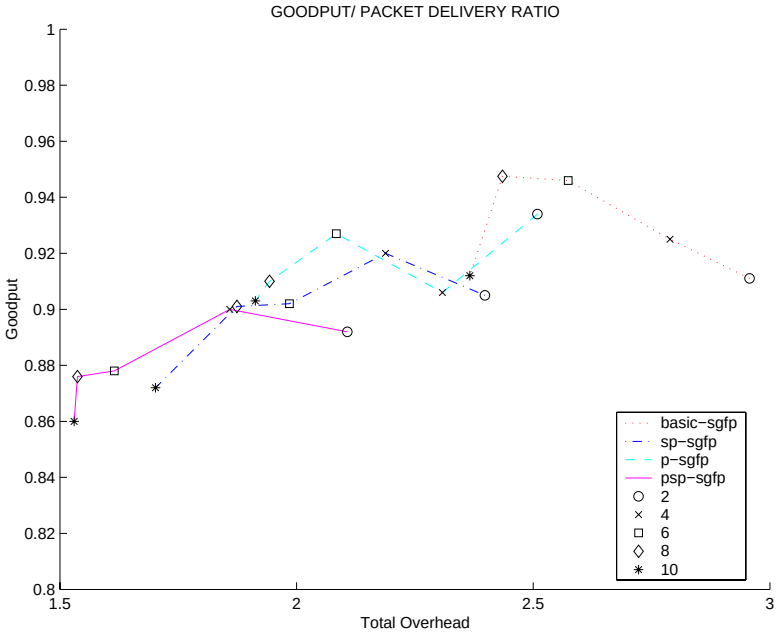

Figure 4.34: Trade-off curve for different refresh intervals

analysis as it is not dependent on the refresh interval. As the refresh interval is increased the frequency of control packet generation is reduced and thus the total overhead is reduced. We can see from the figure that the goodput does not vary appreciably for different values of the refresh interval. Thus by refresh interval values of 6,8,10*secs* are ideal for the source grouped flooding schemes. Again, scheme *psp-sgfp* is the most efficient scheme and the total overhead as low as 1.5 when the refresh interval is 8 or 10 seconds. However, the downside to increasing the refresh interval is the delay in discovering new members. New group members will have to wait for the next route refresh period to respond to the JOIN REQUEST message from the source. Thus if group membership is highly dynamic then a lower value of refresh interval should be used.

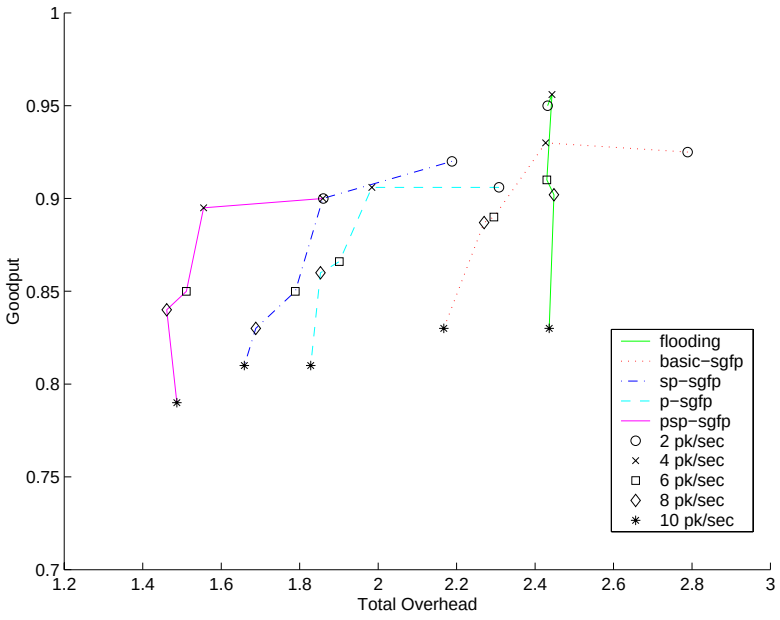

Figure 4.35: Trade-off curve for different traffic load

Figure 4.35 shows the trade-off between goodput and total overhead as the traffic in *packets/sec* is varied. The total overhead for *flooding* remains the same as the traffic load in *packets/sec* is increased. In the case of source grouped flooding schemes, the total overhead initially decreases with increased load and then gradually settles to a constant overhead. Traffic loads of 2-5*packets/sec* seem to be the ideal range for the source grouped schemes. Again the *psp-sgfp* scheme is the most efficient. In fact when the traffic load is 4*packets/sec*, the *psp-sgfp* scheme is 40% more efficient than *flooding*. Also when traffic is very high 10*packets/sec*, the goodput is almost the same for all the schemes. Figure 4.36 shows the trade-off

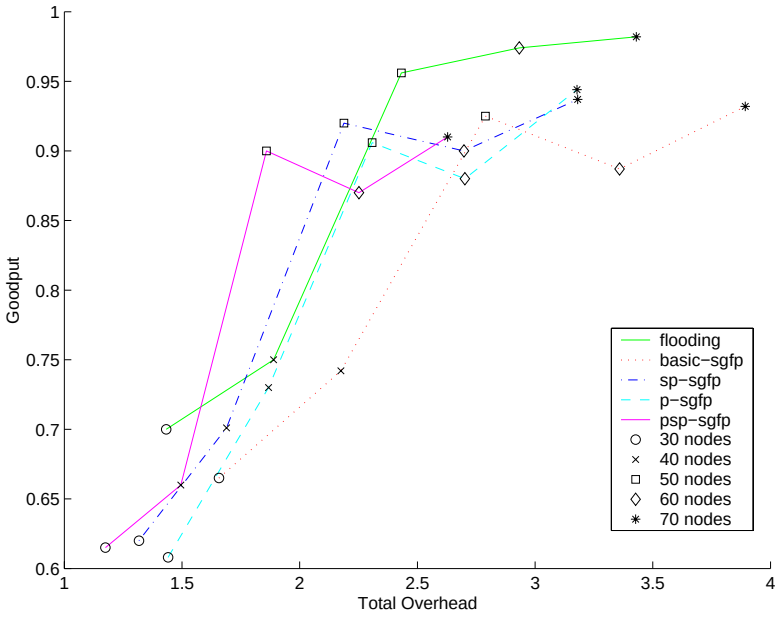

Figure 4.36: Trade-off curve for different network density

between goodput and total overhead as the network density in number of nodes is varied. We can see that all the schemes perform best when there are more than 50

nodes in the network. The *flooding* scheme shows slight increase in performance when the number of nodes in the network is 60 and 70, however the increase in total overhead is appreciable. Scheme *psp-sgfp* is the most efficient of the source grouped schemes and its goodput is within 8% of that of *flooding*. Around 50 nodes in a network of size $1000m$ x $1000m$ seems a good value for the network density.

4.3.9 Some Comments

It should be noted that scenarios with network partitions and persistent mobility, though realistic, represent worst case scenarios. The protocols evaluated should perform better in networks with improved connectivity and predetermined or meaningful mobility patterns. We see that the goodput performance of the source grouped schemes is within 8% of that of *flooding*. Thus the source grouped mesh structure is a stable multicast structure. We also see that, reducing the number of redundant transmissions by using either probabilistic data forwarding (*p-sgfp*) or shortest path flooding groups (*sp-sgfp*), reduces the data overhead while minimally affecting the goodput. Moreover, we see that the shortest path flooding groups described in Section 3.6 are as stable against mobility as the basic flooding groups described in Section 3.1. The benefits of probabilistic data forwarding and shortest path flooding groups are cumulative. This is evident from the performance of scheme *psp-sgfp* which creates shortest path flooding groups and also uses probabilistic data forwarding. Scheme *psp-sgfp* has a goodput performance

within 10% of that of *flooding* for variations of all the simulation attributes. This scheme is the most efficient of all the schemes evaluated. Interestingly, scheme *psp-sgfp* is 25-40% more efficient than *flooding*, considering that JOIN REQUEST messages are periodically flooded in the network. The end-to-end delay of all the schemes is minimal The trade-off curves give us a picture of the operating values for certain key network parameters and protocol parameters.

Chapter 5

Conclusions and Future Work

The inherent constraints of a mobile wireless ad hoc networks viz mobility, bandwidth and energy limitations, pose difficult challenges in designing multicast routing protocols. Thus, a multicast routing protocol for a MANET, should be robust against topology changes and achieve efficient data distribution. In this thesis we presented the source grouped flooding approach to multicast routing in MANETs. The scheme creates flooding groups per source based on distance constraints. The flooding group is a per source, multiple path, mesh structure that is effective and robust against mobility. We demonstrated that using a probabilistic data forwarding mechanism, based on probabilities derived from the network, improves the efficiency of the protocol. Also, the shortest path flooding scheme, improves the efficiency of the protocol due to the reduced number of rebroadcasts. We found that the Probabilistic Shortest Path Source Grouped Flooding Protocol (PSP-SGFP) achieves goodput between 85-91%, within 8% of that of flooding. It is 25-40% more efficient than traditional flooding. We also suggest best performance, operating values for certain network and protocol parameters based on our results. As

an extension to this work we have identified that piggy backing data on the source generated control messages would improve the effectiveness and efficiency of the protocol.

A key lesson learned is that network partitions, and MAC layer contentions and collisions could have catastrophic effects on the performance of the protocols. We identify the following areas of future research:

- Medium Access: Improvements and optimizations in the routing layer may be futile, unless they are working on a reliable MAC layer that scales with the number of nodes.

- Reliable Multicast: True, mesh based protocols with multiple paths are robust against topology changes. However, we need to investigate approaches to achieve total reliability against packet loss.

BIBLIOGRAPHY

[1] B.M.Leiner, D.L.Neison, and F.A.Tobagi. Issues in packet radio network design. *Proceedings of the IEEE, special issue on Packet Radio Networks*, 1987.

[2] J.Jublin and J.D.Tornow. The darpa packet radio network protocol. In *Proceedings of the IEEE*, volume 75, January 1987.

[3] Mobile ad hoc networks ietf chapter
. http://www.ietf.org/html.charters/manet-charter.html.

[4] S.Paul. *MULTICASTING ON THE INTERNET AND ITS APPLICATIONS*. Kluwer Academic Publishers, 1998.

[5] S.E.Deering and D.R.Cheriton. Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, May 1990.

[6] J.Moy. Multicast routing extensions for ospf. *Communications of the ACM*, 1994.

[7] T. Ballardie, P. Francis, and J. Crowcroft. Core based trees (CBT): An architecture for scalable inter-domain multicast routing. In *Proceedings of ACM SIGCOMM*, September 1993.

[8] S. E. Deering, D. Estrin, V. Jacobson D. Farinacci, C.-G. Liu, and L. Wei. The PIM architecture for wide-area multicast routing. *IEEE/ACM Transactions on Networking*, 4(2):153–162, April 1996.

[9] J.Moy. Link state routing. In M.E.Steenstrup, editor, *Routing in Communications Networ*. Prentice Hall, 1995.

[10] G.S.Malkin and M.E.Steenstrup. Distance-vector routing. In M.E.Steenstrup, editor, *Routing in Communications Networ*. Prentice Hall, 1995.

[11] E. Royer and C. E. Perkins. Multicast operation of ad hoc on-demand distance vector routing protocol. In *Proceedings of MobiCom*, Seattle, WA, August 1999.

[12] C. W. Wu and Y. C. Tay. AMRIS: A multicast protocol for ad hoc wireless networks. In *Proceedings of IEEE MILCOM*, Atlantic City, NJ, November 1999.

[13] M. Liu, R. Talpade, A. McAuley, and E. Bommaiah. AMRoute: Ad hoc multicast routing protocol. Technical Report 8, University of Maryland, 1999.

[14] S.-J. Lee, M. Gerla, and C.-C. Chiang. On-demand multicast routing protocol. In *Proceedings of IEEE WCNC*, pages 1298–1304, New Orleans, LA, September 1999.

[15] S. Lee and C. Kim. Neighbor supporting ad hoc multicast routing protocol. In *Proceedings of the ACM/IEEE Workshop on Mobile Ad hoc Networking and Computing (MOBIHOC)*, Boston, MA, August 2000.

[16] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*. IEEE, ieee std 802.11 edition, 1997.

[17] Opnet modeler version 7.0.

[18] E. L. Madruga and J. J. Garcia-Luna-Aceves. Scalable multicasting: The core assisted mesh protocol. *ACM/Baltzer Mobile Network and Applications Journal, Special Issue on Management of Mobility*, 1999.

[19] P. Sinha, R. Sivakumar, and V. Bharghavan. MCEDAR: Multicast core extraction distributed ad hoc routing. In *Proceedings of the Wireless Communications and Networking Conference*, 1999.

[20] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagordia. A performance comparison study of ad hoc wireless multicast protovols. In *Proceedings of IEEE INFOCOM*, Tel Aviv, Israel, March 2000.

[21] J. Lusheng and M. S. Corson. A lightweight adaptive multicast algorithm. In *Proceedings of the IEEE Globecom*, 1998.

[22] C.E.Perkins. *ADHOC NETWORKING*. Addison Wesley, 2001.

[23] S.-J. Lee, W. Su, and M. Gerla. Ad hoc wireless multicast with mobility prediction. In *Proceedings of IEEE ICCCN*, Boston, MA, October 1999.

[24] S.-J. Lee, W. Su, and M. Gerla. *On Demand Multicast Routing Protocol (ODMRP) for Ad hoc Networks*. Internet Draft, draft-ietf-manet-odmrp-01.txt, June 1999. Work in progress.

[25] C.-C.Chiang, M.Gerla, and L.Zhang. Forwarding group multicast protocol (fgmp) for multihop wireless networks. *Baltzer Cluster Computing, special issue on Mobile Computing*, 1998.

[26] S.Murthy and J.J.Garcia-Luna-Aceves. An efficiient routing protocol for wireless networks. *ACM/Baltzer Mobile Networks and Applications, special issue on Routing in Mobile Communications Networks*, October 1996.

[27] C.Ho, K.Obraczka, and G.Tsudik K.Vishwanath. Flooding for reliable multicast in multi-hop ad hoc networks. In *MobiCom Workshop on Discrete Algorithms and Methods for Mobility*.

[28] S.Y.Ni, Y.-C.Tseng, Y.-S.Chen, and J.-P. Sheu. The broadcast problem in a mobile ad hoc network. In *Proceedings of MobiCom*, August 1999.

[29] R. Sivakumar, P. Sinha, and V. Bharghavan. *Core Extraction Distributed Ad hoc Routing (CEDAR) Specification*. Internet Draft, draft-ietf-manet-cedar-spec-00.txt, September 1998. Work in progress.

[30] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector (AODV) routing. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999.

[31] V. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM*, Kobe, Japan, 1997.

[32] C.Ware, T.Wysocki, and J.F.Chicharo. Simulation of capture behaviour in ieee 802.11 radio modems. *Journal of Telecommunications and Information Theory*, 2001.

[33] Billiard mobility

. http://w3.antd.nist.gov/wctg/manet/prd_aodvfiles.html.

[34] Quality of service requirements for multimedia applications

. http://www.doc.ic.ac.uk/ ejhp98/article1/.